1-31-1993

A Sign-to-Speech Translation System

Koka Veera Raghava Rao
University of Nebraska-Lincoln

A SIGN-TO-SPEECH TRANSLATION SYSTEM

by

Koka Veera Raghava Rao

A THESIS

Presented to the Faculty of

The Graduate College at the University of Nebraska

In Partial Fulfilment of Requirements

For the Degree of Master of Science

Major: Computer Science

Under the Supervision of Professor Peter Z. Revesz

Lincoln, Nebraska

January, 1993

A SIGN-TO-SPEECH TRANSLATION SYSTEM

Koka Veera Raghava Rao, M.S.

University of Nebraska, 1993

Adviser: Peter Z. Revesz

This thesis describes sign-to-speech translation using neural networks. Sign language translation is an interesting but difficult problem for which neural network techniques seem promising because of their ability to adjust to the user's hand movements, which is not possible to do by most other techniques. However, even using neural networks and artificial sign languages, the translation is hard, and the best-known system, that of Fels & Hinton (1993), is capable of translating only 66 root words and 203 words including their conjugations. This research improves their results to 790 root signs and 2718 words including their conjugations while preserving a high accuracy (i.e., over 93 %) in translation. The use of matcher neural networks (Revesz 1989, 1990) and asymmetric Hamming distances are the key sources of improvement. This research aims at providing a means of communication for deaf people.

ACKNOWLEDGMENTS

I would like to express my utmost gratitude to my advisor, Dr.Peter Revesz, for his help, guidance, time and encouragement, and most of all his patience, during the long, arduous process of producing this thesis. Without Dr.Revesz's ideas and support, the research in this thesis would not have taken place.

I would like to thank the committee members Dr. Peter Revesz, Dr. Ashok Samal and Dr. Prabir Bhattacharya for their comments and suggestions.

Finally, I wish to thank my wife, Madhavi for her support and help in hard times.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Sign languages have been used for a long time. In any sign language, information is transmitted through manual/visual channels instead of the oral/auditory channels used for spoken languages. There are many deaf people in the United States for whom manual communication is the primary mode of communication (Mayberry, 1978).

Sign language was a common and universal means of communication throughout the Great Plains among the various American Indian tribes who spoke different vocal languages (Tomkins 1969). Today sign languages form a very important group of languages, with over a hundred thousand people worldwide who use them daily as their primary mode of communication.

When a signer needs to communicate with a non-signer,1 some form of interpretation is necessary. Interpretation can be provided through anything from pencil and paper to hiring a certified interpreter. The written approach is very slow and often frustrating. Hiring interpreters also has drawbacks; it is expensive; an interpreter is not always available; and some people are uncomfortable speaking through an interpreter because it seems less personal and less private.

One possible way of improving this situation would be to automate the interpretation process. Computer recognition techniques could be designed to translate the signers handshapes and movements into spoken or written English. These techniques could then be used to enhance or replace human interpretation.

1.1 American Sign Language

American Sign Language (ASL) is a complete and well-formed language developed naturally within the American deaf community. Its grammar is quite distinct from that of English or any other spoken language (Wilbur 1979, Meier 1991). It is the fourth most common language used in the United States.

Many people mistakenly believe that American Sign Language(ASL) is English conveyed through signs. Some think that it is a manual code for English, that it can express only concrete information, or that there is one universal sign language used by deaf people around the world.

Linguistic research demonstrates, however, that ASL is comparable in complexity and expressiveness to spoken languages. It is not a form of English. It has its own distinct grammatical structure, which must be mastered in the same way as the grammar of any other language. ASL differs from spoken languages in that it is visual rather than auditory and is composed of precise handshapes and movements.

ASL is capable of conveying subtle, complex, and abstract ideas. Signers can discuss philosophy, literature, or politics as well as football, cars or income taxes.

Sign Language can express poetry as poignantly as can any spoken language and can communicate humor, wit, and satire just as bitingly. As in other languages, new vocabulary items are constantly being introduced by the community in response to cultural and technological change.

ASL is not universal. Just as hearing people in different countries speak different languages, so do deaf people around the world sign different languages. Deaf people in Mexico use a different sign language from that used in the U.S. Because of historical circumstances, contemporary ASL is more like French Sign Language than like British Sign Language. ASL was developed by American deaf people to communicate with each other and has existed as long as there have been deaf Americans. ASL is now used by approximately one-half million deaf people in the U.S. and Canada.

Apart from ASL, the Indian sign language is the world's most easily learned language because it is elemental, basic, logical and the signs in general are what should properly be made to illustrate the idea - the language being largely idiomatic - conveying ideas (Tomkins 1969).

1.2 Automatic Recognition

While there have been many studies on ASL (Poizner 1983, Poizner et al. 1987, Stungis 1981, Loomis et al. 1983) including some work towards an automatic transla-

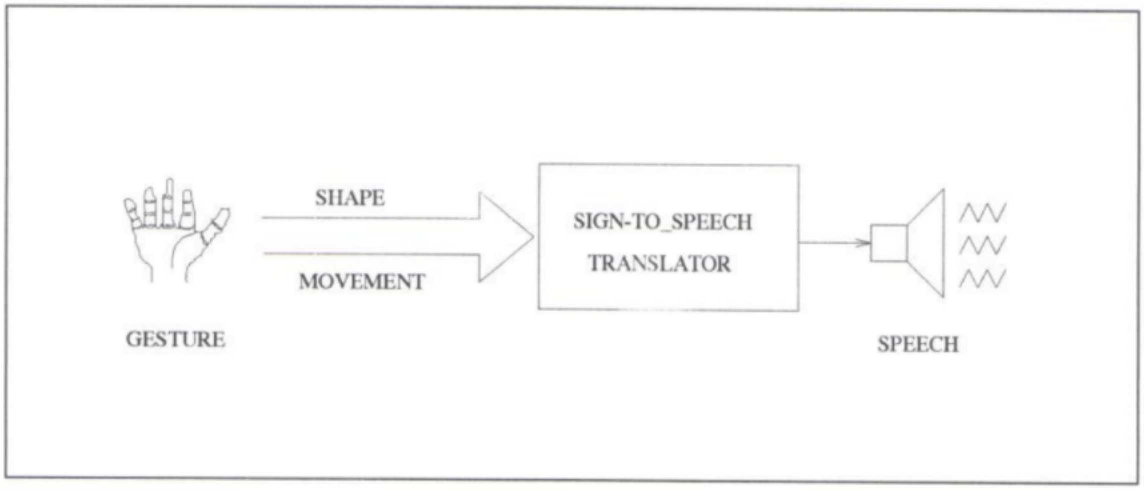

Figure 1.1: Block Diagram of Sign-to-Speech Translation

tion to English, it turns out to be a very complex language to study directly. Instead of attacking the problem of ASL head on, some authors considered the more manageable problem of translating various artificial sign languages to English. Even that is very difficult. Recently, only a few projects have been directed towards this problem.

Figure 1.1 shows the simplified block diagram of an automatic recognition system which senses the users hand shape and movement and translates it to speech. Researchers have investigated different types of translation techniques. Some of them used vision based techniques based on markers to capture the users hand shape and movements. Others used DataGlove or CyberGlove that are equipped with fiber optic sensors which measure the angle of flexure at each joint.

Each of the above techniques have their own advantages and disadvantages. Vision based techniques are much slower than sensor based techniques as they have to process more data. Hence for real-time applications sensor based techniques are preferred over vision based techniques. however for off-line analysis vision based techniques may yield better performance than sensor based techniques.

Tamura and Kawasaki (1988) presented a system that recognized a small list of handshapes from a videotape of a single signer. This system was designed to recognize only 20 signs and had only a 45 % accuracy rate. Moreover, the system could not handle many signs without losing accuracy.

Another automated interpretation system was presented by Kramer & Leifer (1987, 1989) as part of a computer-based system that recognized American fingerspelling and returned a spoken English translation. In this system, a signer would be able to communicate with a non-signer by manually spelling English words while wearing a CyberGlove. Computer recognition algorithms were used to interpret the information from the glove, and the resulting English text was converted to synthetic speech for the non-signer to hear.

The recognition algorithm in Kramer's system had some limitations. First, the algorithm expected a stop in the hand motion after each letter to trigger the recognition process. The requirement to stop between letters is undesirable since fluent fingerspellers use nearly continuous motion in producing a sequence of letters (Guillory, 1966). Further, it was designed to recognize static hand shapes only so each letter had to be produced as it were in isolation.

Glove-Talk (Fels, 1990; Fels & Hinton, 1993) was designed to recognize handshapes and movements using a DataGlove as the input device. The DataGlove is a nylon glove with sensors designed to read finger angles and hand locations. The Glove-Talk system used neural networks to perform the recognition and had a reported accuracy rate of 92 %. However, a drawback to this system is that its design could handle only 203 words with conjugations. Also, the sign system would have to be be completely redesigned to handle an increased vocabulary with the same accuracy.

Another important direction in this research is to study the co-articulation affects prevalent in fingerspelling and automatic recognition algorithms. Shirley (1992) examined the range and extent of co-articulation in American fingerspelling and its implications for automatic sign recognition. She concluded that recognition algorithms can be improved by taking co-articulation into account and by concentrating on just the selected features of handshapes rather than on full handshapes since less co-articulation may be present for selected features. She also concluded that the context of the previous sign should always be taken into account when the next sign is being processed in the sequence.

The best results to date are those of Fels and Hinton (1993) who achieved a robust translation of 66 root signs plus their conjugations. Of course, this falls short of an average English vocabulary, which has over 20,000 words, and even of the basic English vocabulary, which contains about 850 words (Ogden 1968). Their important and pioneering study however leaves much room for improvement.

The research presented here improves the previous translation works using artificial sign languages. In particular an interface was designed, which robustly translates 2718 gestures to spoken English words and which is capable of adapting to different users with different hand structures. The results obtained show that the accuracy of the translation system remains quite high, i.e. 93 %. This can be considered a very high accuracy, considering the complexity that the system has to deal with in providing an adaptive interface between a user and a text-to-speech synthesizer.

To provide such an interface neural networks seems necessary, and indeed they form the kernel of the translation system designed. The particular neural networks used are adapted from Revesz (1989, 1990) are not as well-known as the backpropagation algorithm (Rumelhart et al. 1986) used by Fels and Hinton (1993) or the sparse distributed memories of Kanerva (1988) to which they seem closest. Therefore, the research is also novel in presenting an application of a less-known biologically motivated network.

This report is organized as follows. Chapter 2 describes some of the possible sign-to-speech models. Chapter 3 describes an overview of the sign-to-speech system. Chapter 4 describes the basic concepts of Asymmetric Hamming distance and Matcher neural networks. Chapter 5 describes the computerized generation of hand symbols. Chapter 6 describes the results obtained, conclusions and future work.

Chapter 2

Translation Models

2.1 Overview

The task of converting hand movements to speech can be accomplished in many ways. The user makes a hand gesture which is recorded by the recognition system. Some of the salient features from the recorded data are extracted and fed as an input to a neural network. The neural network associates the input with a sound description which is then converted to speech by the speech synthesizer. The key part of this system is the mapping of hand data to the synthesizer input. There is a multitude of possible models to choose from, which form a spectrum based on the granularity of speech.

Each of the sign-to-speech models can be compared using the properties vocabulary size, initial mapping complexity, hand movement speed, user learning complexity and sound production lag time. The vocabulary size is the number of different words possible with each particular model. Initial mapping complexity is a relative measure of the difficulty of defining the sign-to-speech mapping. The fewer number of output categories, the easier it is to define an initial mapping.

Hand movement duration is defined as the approximate amount of time a user

Table 2.1: Sign-to-Speech Models and their properties

Models	Vocabulary	Mapping	Motion (msec)	Learning	Response
Artificial Vocal Tract	unlimited	hard	10	hard	FPD
Phoneme Generator	unlimited	medium (≈ 50)	50	medium	$2 \times$ FPD
Finger Spelling	unlimited	easy (≈ 26)	80	medium	$(\lvert\text{word}\rvert + 1) \times$ FPD
Syllable Generator	depends on syllables	number of syllables	200	medium	FPD
Word Generator	number of words	(≈ 850) for Basic English	500	easy	FPD

spends in making successive sounds, without making the speech output sound disjointed. User learning complexity is a measure of the relative difficulty for a user to learn to speak (based on the initial mapping). Normally this measure is a combination of mapping complexity and hand movement duration. System response time is the amount of time lag from the gesture to speech conversion. It is expressed in terms of the forward pass delay (FPD) of the translation system.

Each of the various sign-to-speech models possible as illustrated in Table 2.1 are discussed in the following sections. After examining the merits and demerits of all the models carefully, this research considers the hand as a word generator model for the implementation of the sign-to-speech translation system.

2.2 Hand as an Artificial Vocal Tract

In this model the hand shape and movements mimic the vibration of articulators in the vocal tract. Once all the possible sounds of the vocal tract and co-articulation effects are represented by hand movements, any human sound can be produced. Thus this model has unlimited vocabulary size and natural sounding speech. The mapping from the hand gestures to speech in this model is complex because of the fact that each place of articulation must have an associated hand position. As well, any possible movements of the articulators must be mapped to the movements of the hand shape.

The second difficulty arises in the sound production time limit. For connected, intelligible speech, it is important to note that most sounds do not extend more than about 20 msec. Basically this means that the duration of the user's hand movements must be of the same order of magnitude as the articulators. The translation system should also operate within this time frame. Hence learning to speak with this model is a very difficult task.

2.3 Hand as a Phoneme Generator

There are 45 to 48 different phonemes in English (Shriberg and Kent 1982). In this sign to speech model, hand movements and shapes represent phonemes. With all the phonemes represented a user has an unlimited vocabulary; any word can be phonetically spelled out. In addition, as all the possible word sounds are available the resultant speech sounds almost natural. Only 45 to 48 hand gestures are necessary to define the initial mapping. However, care must be taken to make frequent pairing of phonemes close in terms of hand movements. Considering the above points, the definition of the initial mapping is probably medium hard on the relative scale of difficulty.

For continuous speech the phonemes must be produced in quick succession. Consider a phoneme duration of 50 msec. Once the gestures for this phoneme is made, the user has approximately 50 msec to make the next movement, otherwise there will be a period of silence between successive phonemes. For a user to learn 45-48 unique hand positions is not too difficult; however, to learn to make them quickly enough for continuous sounding speech is very difficult. For these reasons the task of learning to speak with the hand as phoneme generator model is medium hard.

2.4 Hand as a Letter Generator (Fingerspelling)

In this model, each hand gesture is mapped to a letter. Once all the letters in English are represented, the user has unlimited vocabulary. Unfortunately, the sound quality is very poor. There is no account of the variation in word sound since each letter is made individually. There is no global word property to give tailored sounds to speech. Combining the simplicity of the initial mapping with the hand movement speed makes this model medium hard for the user to learn.

Apart from ASL, there are many manual systems based on spoken language. Some used in the United States includes Signed Exact English (SEE), Manual Coded English (MCE), and Pidgin Signed English (PSE). In all of these manual communication systems, including ASL, 'fingerspelling' based on a manual alphabet, is used to spell a word for which there is no specific sign. Fingerspelling is thus commonly used to communicate names, addresses, and new or technical words that are not in the sign lexicon.

2.5 Hand as a Syllable Generator

In this model, a hand gesture is mapped to a syllable. Sequences of gestures form words. The major drawback of this system is the prohibitively large number of syllables in English. A subset of all the syllables must be used. The vocabulary size is then restricted to the meaningful permutations of these syllables. Clearly, as the number of syllables increases, it becomes more difficult to define an initial mapping to convert gestures to speech.

As in fingerspelling, each word must be syllabically spelled out. However, there are two main differences. First, the number of syllables in a word is less than the number of letters. Second, each syllable can be spoken in isolation. For these reasons, the hand movement speed is only the duration of a syllable ($\approx$ 200msec). The fact that syllables can be said in isolation also means that the sound production lag is only the forward delay through mapping.

2.6 Hand as a Word Generator

In this model, hand gestures are mapped to individual words. Vocabulary size is just the number of words chosen. In English there are more than 20,000 words. As in the case of the previous two models, if all the English words are represented by gestures, defining the initial mapping will be extremely difficult. Instead a subset of English language may be used. Fortunately, a language called Basic English (Ogden 1968) has been devised which has only 850 English words. These 850 words do all the essential work of 20,000 words of English (Ogden 1968). Thus, with hand as a word generator model, a vocabulary of 850 words is sufficient for a reasonable level of conversation.

To create an initial mapping for this size of vocabulary is not too difficult, compared to some of the other models. In addition, many hand symbols have been defined in ASL which can be used as a guide in designing initial mapping. Further, as many of the words in Basic English are concrete (pictured) objects, gestures can be made to resemble the objects they represent. The above points make designing the initial mapping relatively easy as well as making the user's job of learning to speak simpler.

The allowable hand movement duration is the duration of the previous word spoken. This is typically in the 500 msec range. A further feature of the hand as word generator model is that each word can be said in isolation; thus, the sound production lag is only the forward mapping delay.

2.7 Selecting a Model for the Translation System

In this research, hand as a word generator model was chosen to build the sign-to-speech translation system. The hand as a word generator model has several important advantages over the other models as already seen in section 2.6. In addition, once developed, a useful system results. Using Basic English as the available vocabulary a ceiling of 850 words is set. With this vocabulary and the system designed, a person should be able to learn to converse at a moderately acceptable level.

Another advantage is the amount of time allowed to generate a word. To avoid gaps, the duration of each previous word spoken limits the production time of the desired current word; assume for example this time is 500 msec. The user's gesture creation time plus the neural network forward propagation must then be less than 500 msec for the speech to sound connected. This duration is much longer than any of the other models. The major drawback with this model is that the speech quality is not as good as with some other models, due to the lack of control of interword speech parameters. The next chapter describes the implementation of the hand as a word generator model.

Chapter 3

Sign-to-Speech Translation

3.1 Overview

The purpose of the sign-to-speech system is to convert user's hand gestures to speech based on a sign-to-word model where each gesture is mapped to an English word. The basic operation of the system is as follows: the user forms a hand shape and makes a movement forward and back in one of the six possible directions. The system records the user's hand shape and the direction of movement. The direction of movements recognized by the system include right, left, upward, downward, away and towards the user. The hand shape and direction of movement are then mapped to an English word.

The sign-to-speech translation system is equipped with CyberGlove which monitors the user's hand shape, Flock-of-Birds which senses the user's hand movements and a DECtalk speech synthesis system. The system was designed on a SUN SPARC workstation under UNIX environment.

This chapter is organized as follows: Section 3.2 describes the complex structure of the hand. Section 3.3 thru section 3.5 describes the hardware used and the nature of measurements made. Section 3.6 describes the sign-to-speech translation system designed. Section 3.7 describes the performance of the strobe detector which is one of the critical building block in the sign-to-speech translation system designed.

3.2 Hand Structure

The human hand is a complex structure with several kinds of bones that are interconnected by various joints having different degrees of freedom. This research considers 17 different joint movements as described below:

The first group of joints considered are the **Metacarpophalangeal Joints (MCPs)**. These joints represent the knuckles of the hand. They are condyloid joints with 2 degrees of freedom. The flexion or extension of these joints up to 100° occurs about a transverse axis of the hand. An abduction or adduction of these joints - as much as 30° in the case of the index finger - occurs about an anteroposterior axis with reference to the middle finger.

Second group of joints considered are the **Interphalangeal Joints**. These are hinge joints with 1 degree of freedom. Flexion or extension of these joints up to 90° occurs about a transverse axis as in the case of MCPs. The Proximal Interphalangeal joints (PIPs) are between the proximal and the middle phalanges. (The Distal Interphalangeal joints (DIPs) are between the middle and the distal phalanges; however, the movement of these joints are not considered).

Third group of joints considered are the **First Carpometacarpal Joint** of the thumb. This is a saddle shaped joint with 2 degrees of freedom between the trapezium and the first metacarpal bones. Flexion or extension of this thumb joint occurs about a plane which is approximately 60° to that of the hand. Flexion brings the thumb ventrally to the plane of the hand towards the palm, while extension brings the thumb back into the plane of the hand. Abduction and adduction of the thumb occurs about an axis perpendicular to the plane of the hand. Abduction is the movement of the extended thumb away from the index finger, while adduction brings the extended thumb against the index finger.

The fourth and last group of joints considered are the **Wrist Joints** which are located between the distal end of the radius and the ulna and the proximal row of carpal bones. Flexion and extension also known as wrist pitch occurs about a transverse axis. Abduction and adduction also known as wrist yaw occurs about an anteroposterior axis. The next section gives a description of the hardware used to measure the angles at each of the joints mentioned above.

3.3 CyberGlove

The Virtual Technologies *CyberGloveTM* (Model Number CG1801) is an instrument capable of measuring the movements of the user's fingers and hand. It is used to record the shape of the user's hand. It is provided with a glove equipped with fiberoptic sensors for the user to wear and is connected to the CyberGlove Interface Unit (CGIU). The CGIU contains the controller which manages the protocol between the host workstation and the glove. The CGIU is connected to one of the serial (RS232C) ports of the SUN SPARC workstation.

Although the CyberGlove (Model Number CG1801) is capable of measuring 18-joint movements, the present research considers only 17-joint movements. The position of the CyberGlove sensors considered are illustrated in the Figure 3.1. Five of these measure the flexion or extension of the MCPs, five for the flexion or extension of PIPs, four for the abduction or adduction between adjacent MCPs, one for thumb rotation, and one each for wrist pitch and wrist yaw.

The CyberGlove sensors give digitized output values that vary linearly with the angle of the joints over which they are located. Therefore these sensors can easily monitor the movement of the joints described in Section 3.2. MCP, PIP and wrist pitch angles are defined such that joint flexure corresponds to increasing A/D values returned from the CGIU. Abduction sensors are defined such that finger spreading produces decreasing A/D values. Flexing of the wrist joint to the side of the pinkie finger corresponds to increasing sensor A/D values.

The software configures the CyberGlove to operate at 9600 baud rate. The response time of the CyberGlove is 15 msec. That means it takes about 15 msec for the host to receive the data once it is requested.

3.4 Flock of Birds

Ascension Technology Corporation's *FlockofbirdsTAI* is a position sensor that is used to track the user's hand position and orientation over short ranges. Other applications include: head tracking in flight simulators/trainers and virtual reality games, real-time control of 3D images in computer graphics workstation, 3D measurement of medical instruments, biomechanical measurement of anatomical parts, manipulation

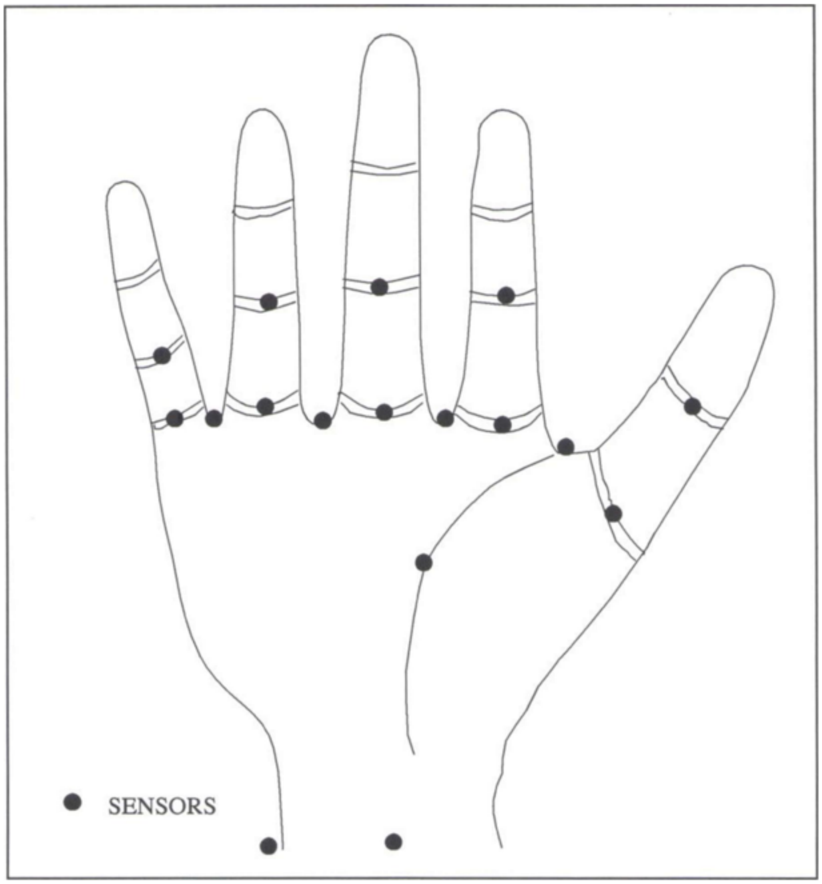

Figure 3.1: CyberGlove sensor location

of telerobotic controls, and real time interaction with virtual images.

It has a transmitter and a receiver. The transmitter generates a magnetic field whose field strength is measured by the receiver in order to determine its 3D position (X,Y and Z coordinates) with respect to the transmitter. The flock of birds also measures the orientation angles of the receiver with respect to the transmitter. The orientation angles are defined as rotations about the X, Y and Z axes of the receiver. They are called in the nomenclature as roll, elevation, and azimuth respectively. This research uses only X, Y , Z coordinates and roll for tracking the user's hand move-

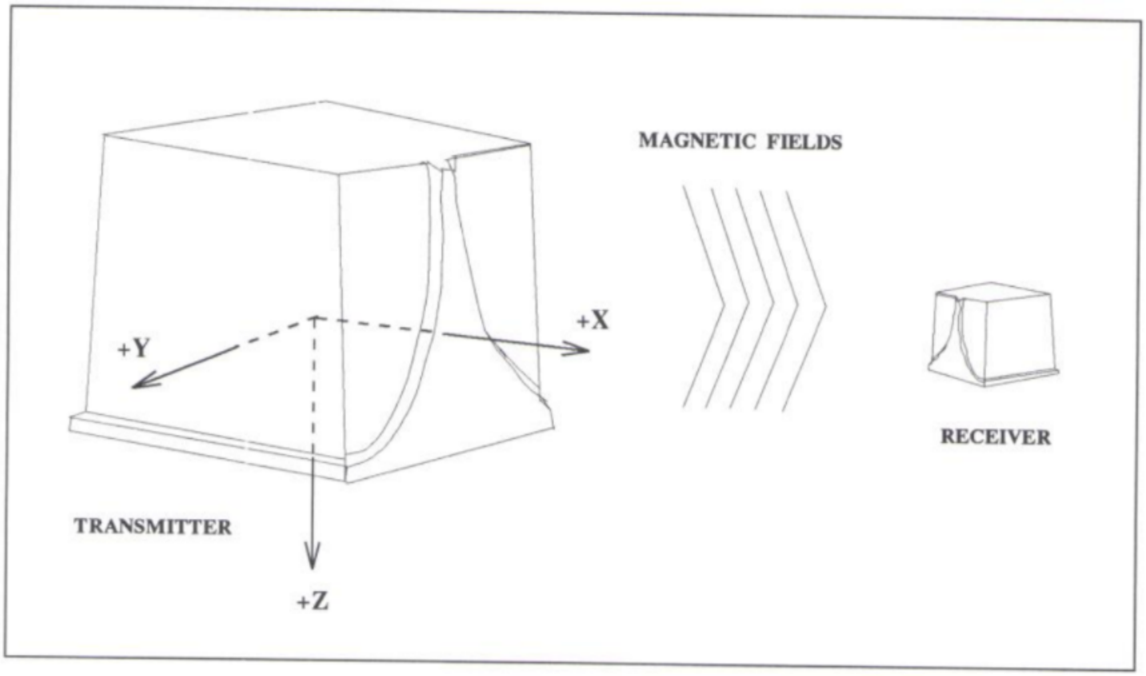

Figure 3.2: Flock of birds operation

ments.

The Flock of birds controller is connected to one of the serial ports (RS232C) of the SUN SPARC workstation. The software configures the Flock of birds to operate at 9600 baud rate with a capability to make 100 measurements per second. The Flock of birds is capable of tracking multiple receivers. It can handle upto eight transmitters and receivers. The present system uses only one receiver and one transmitter as this research considers only one-handed signs. However, in future it can be upgraded to work with sign languages that use both hands.

3.5 DECtalk

DECtalk^{TM} (Model Number DTC01) is a text-to-speech converter designed by Digital Equipment Corporation that provides any computer with a human-sounding voice. Figure 3.3 shows the various modules used in the DECtalk speech synthesis system. It has sentence parser which breaks up the input stream into separate words and locates some clause boundaries (such as commas and other punctuation marks). A word parser breaks compound words into their component parts, yielding words in their final pronounceable form. A dictionary manager searches the pronunciation dictionaries which are built into DECtalk. A phrase structure module recombines all phonemic output from the dictionary search and other modules. A phoneme-to-voice module processes clauses passed from the phrase structure module and converts them to control signals for the speech synthesizer. The digital speech synthesizer computes the speech waveform with acoustic characteristics that are determined by the synthesizer control commands received.

It has a large selection of user controllable speech parameters. For example, the user can control the speaking rate and word stress by sending some control characters to the DECtalk. For words not in the DECtalk dictionary, there is a letter-to-sound module which uses a set of English pronunciation rules to assign phonemic form and lexical stress patterns to words not found in the dictionary. However if the user wants to provide his own pronunciation he can break up the word into phonemic code which is accepted by DECtalk. This research uses only the DECtalk's builtin dictionary and does not provide control over speech parameters such as word stress and word rate. However, it can be integrated into the system in future by making some design changes.

The software configures the DECtalk to operate at 9600 baud rate. It is connected to one of the serial (RS232C) ports of the SUN SPARC workstation. DECtalk reads text at about 180 words per minute. This is a typical reading rate that is comfortable for most situations. Words per minute is a relative measurement. DECtalk adjusts its speech rate for average-length words. It does not count the number of words in a sentence before speaking them. Sentences with long words take longer to speak than sentences with short words

3.6 System operation and implementation

The sign-to-speech system designed is illustrated in the Figure 3.4. The initializer module opens up the serial communication ports on the SUN SPARC workstation and initializes the protocol associated with the three devices. The flock of birds starts recording the X,Y and Z position of the hand and stores their values in a buffer of length N. At any given time T the buffer stores data samples from T to T-N-1.

The strobe detector module calculates the displacement in the hand position by inspecting all the data values in the buffer. Once it has detected a displacement 'd' between any two values in the buffer it records a strobe and triggers the CyberGlove with a data request. The strobe detector then clears all the data values in the buffer and replaces them with the most recent value. This is done in order to prevent any false detection of the strobe in the future. The strobe detector also memorizes the most recent direction in which the user moved his hand. Then it expects the user to move the hand in the opposite direction to the one it memorized. Once it detects the displacement of the user's hand in the opposite direction it recognizes the completion

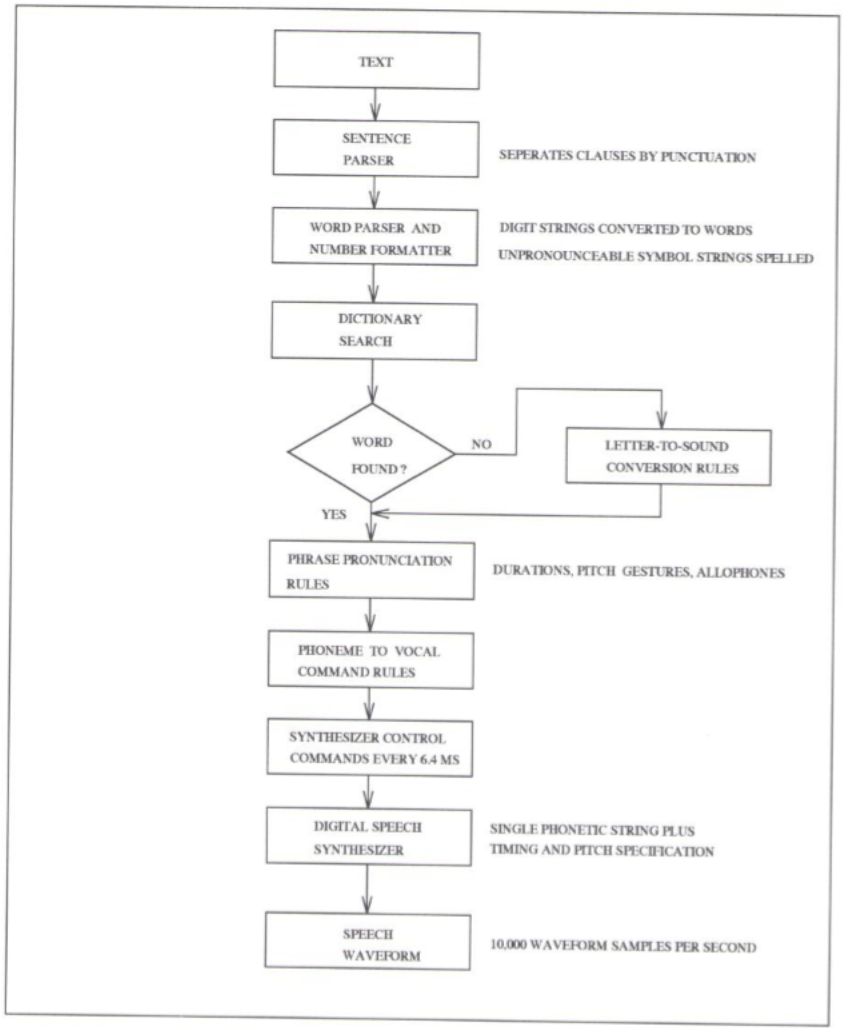

Figure 3.3: Text-To-Speech Conversion

of the gesture made by the user.

The strobe detector then clears its memory and also the buffer and the cycle repeats. The timer/counter module helps the system to forget the past information after N cycles and updates the system with the most recent data. The counter is incremented when ever the strobe detector fails to record a strobe and is reset to zero when the strobe detector detects a strobe. When the timer/counter value in N it clears the buffer and also the memory associated with the strobe detector. It then resets itself to zero.

The CyberGlove data is fed to a feature extractor after the gesture has been recognized by the strobe detector. The feature extractor then extracts the salient features from the raw data and encodes them into a 17 bit binary pattern. The features extracted are fed to a neural network which detects and corrects some of the errors committed by the user. The roll detector module detects the rotation of the user's hand. The strobe detector after recognizing the gesture detects the direction in which the user moved his hand.

The extracted data consisting of 17 bit CyberGolve data and 7 bit Flock of birds data are fed as input to the feature to word associator which associates the input with an English word. The word is then sent to the DECtalk which speaks out the word. The next chapter describes in detail the feature lists and the neural networks used.

3.7 Strobe detector performance

The performance of the strobe detector is critical for the working of the sign-to-speech translation system. The most important factor that should be considered in the design is the sensitivity of the strobe detector to the user's hand movements. If it is too sensitive, then the frequency of false detection increases because it responds to slight movements of the hand. If it is less sensitive it results in a miss as it may not capture some of the intended user hand movements.

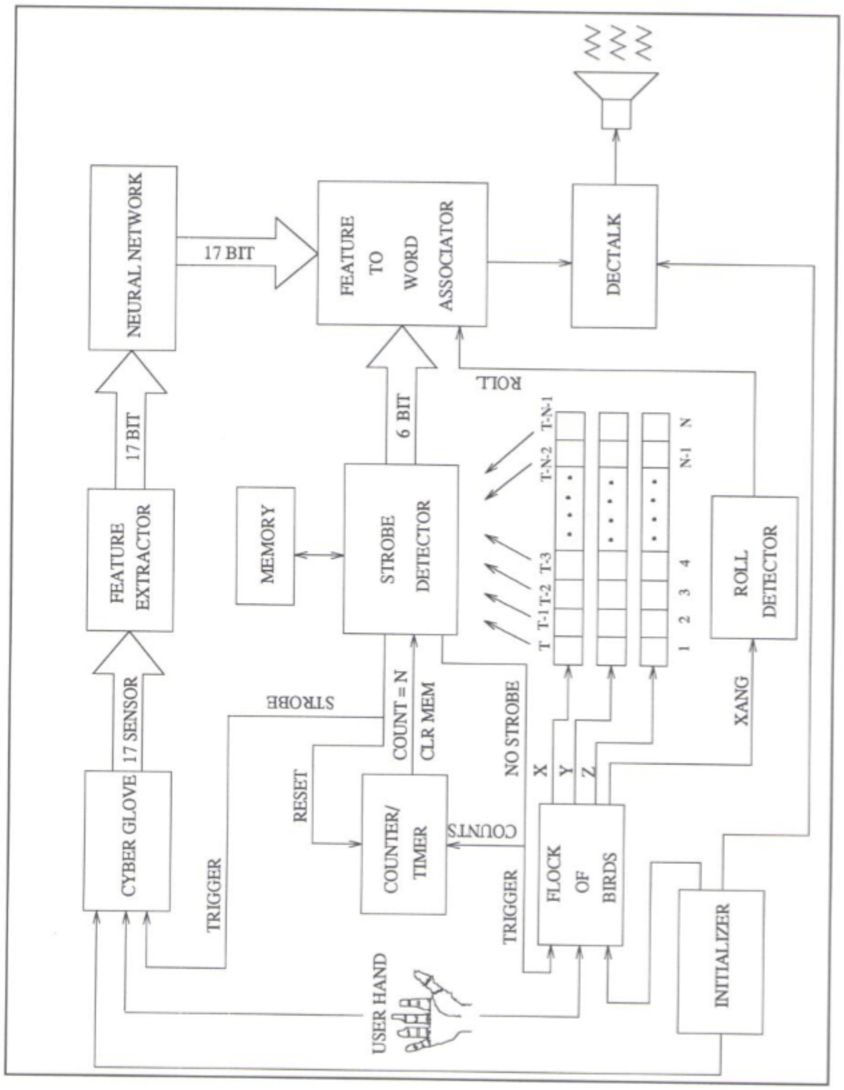

Figure 3.4: Sign-to-Speech translation system

The design was aimed at minimizing the errors due to false detections and misses. The system does not restrict the user to work with in a particular reference frame, which means that it does not fix the coordinates of the initial position. The advantage of this approach is that the user can make a gesture from any point in the 3D space provided it is within the range of the magnetic field generated by the Flock of the birds. This approach eliminates the possibility of an error that may result due to improper positioning of the initial position.

The movement of the hand in the opposite direction during any gesture reduces the probability of error due to false detection. In order to understand how the this approach reduces the probability of error due to false detection let us consider a design where the user need not make the opposite movement. When the user makes the one directional gestures he has to keep track of the amount of displacement needed mentally. If he exceeds the displacement there is a chance that the strobe is detected more than once. However in the bi-directional gestures there is no chance of a strobe being detected more than once due to the hand movements in opposite directions. Hence the reduction in the probability of error due to false detection.

Following are the three parameters that must be optimized in the present design in order to get the best performance from the strobe detector:

1. **Buffer Length**: The buffer length N is one of the critical parameter that need to be optimized in order to get the best performance from the strobe detector. If N is too long then the user's intentions may not match the systems interpretation

because of the excessive memory. The advantage of having a large N is that the system tracks the users hand movements more closely. However it increases the system response time and an error committed by the careless user will get multiplied due to excessive memory. Thus increasing the probability of error due to false detection. If the buffer length is too small then the system may not capture the users intentions. Moreover the user has to make gestures at high speed to make the system record his gesture. In this case the probability of the occurrence of a miss is higher. After considering all the above facts the present system was designed with a buffer length of 50.

2. **Displacement**: It is another factor that must be studied closely in order to design a system that is user friendly. The physical restrictions on the user's hand movements is one of the factors considered in choosing an optimal value for the displacement. If the displacement is large then the probability of the occurrence of a miss is larger. If the displacement is small then the probability of false detection is more. An optimal value of 3 inches was chosen in the present system to maximize the performance.

3. **Maximum Timer Count**: The maximum timer count was found to have the same effects as that of the buffer length. So in the present system the maximum timer count was set to N in order to maximize performance.

Considering all the above factors the strobe detector performance was very high(99 %). The next chapter discusses the common type of errors associated with the CyberGlove environment and how they are corrected to improve the overall system performance.

Chapter 4

Feature Extraction and Neural Networks

4.1 Overview

Feature systems for signs are less well established than for spoken languages. In fact, the distinctive feature list differs with each researcher.

Woodward (1973) used a binary feature system based on articulatory characteristics to describe the handshapes used in ASL. The system included a binary feature for each finger: [thumb], [fore], [middle], [ring], [pinky] and five binary features for describing the full handshape: [closed], [spread], [bent], [contact] and [crossed]. With the ten features, specified either positive or negative, 40 separate ASL handshapes can be uniquely described.

Mandel (1989) devised a feature set for the linguistic description of ASL. He argued that all of the fingers of any given sign are divided into two groups: foreground (the selected group, or the fingers necessary for the recognition of the letter) and background (the unselected group). The selected fingers can be in any position except closed and must all be in the same position. All unselected fingers must also be in the same position, but can only be either fully open (straight) or fully closed against the palm. This feature analysis reduced the number of possible feature combinations within a handshape. This helped to reduce redundancy in writing phonological rules by referring to all the fingers as two groups instead of individually.

4.2 Feature Lists

This research will use a special type of feature lists to describe signs. More specifically, it will use joint feature lists. That is, it takes the feature list of each sign to be a pattern of 0's and l's where at each position a 1 signifies the bending or a 0 signifies the straightening of a certain joint.

This research is not using the angle values directly as provided by the CyberGlove. It converts each angle value into a "1" if the joint is bent or into a "0" if the joint is straightened. To do the conversions to binary, some threshold values are used that have to be adjusted separately for each user.

Table 4.1 shows a simplified example of joint feature lists considering only the five PIP joints. There the representations of the signs for "is", "a", "there", "she", "he", and "woman" are listed. For example, the word "there" is represented by the pattern 00111 which signifies that while making the sign for this word we have to straighten the thumb and the index PIP joints and bend the other three. This example will be elaborated further in the following sections.

Table 4.1: A simple example using PIP feature lists

	FEATURES				
SIGNS	**TP**	**IP**	**MP**	**RP**	**LP**
IS	1	1	1	1	1
A	1	1	1	0	0
THERE	0	0	1	1	1
SHE	0	1	0	1	0
HE	1	0	1	0	1
WOMAN	0	0	1	0	0

4.3 Asymmetric Hamming Distance

Joint feature lists are the primary means of distinguishing between different hand signs. Intuitively the feature lists are required to be as different as possible so that minor errors of one or two bits do not lead to confusion among different signs.

The difference between two feature lists can be measured in several ways. The most common mathematical measure for the dissimilarity between two binary patterns is the Hamming distance. The Hamming distance is the number of corresponding bits in which the two patterns differ. For example, the Hamming distance between 00111 and 01010 is three.

The Hamming distance is widely used in digital communication problems where there is an equal probability of 1 getting corrupted to 0, and 0 getting corrupted to 1 during signal transmission. Such a type of error is known as symmetric error as shown in the Figure 4.1. The other type of error known as asymmetric or unidirectional error occurs when the probability of 1 getting corrupted to 0 is much higher than the probability of 0 getting corrupted to 1 as shown in the Figure 4.1 or vice versa. These two types of asymmetric errors are also referred to as omission and

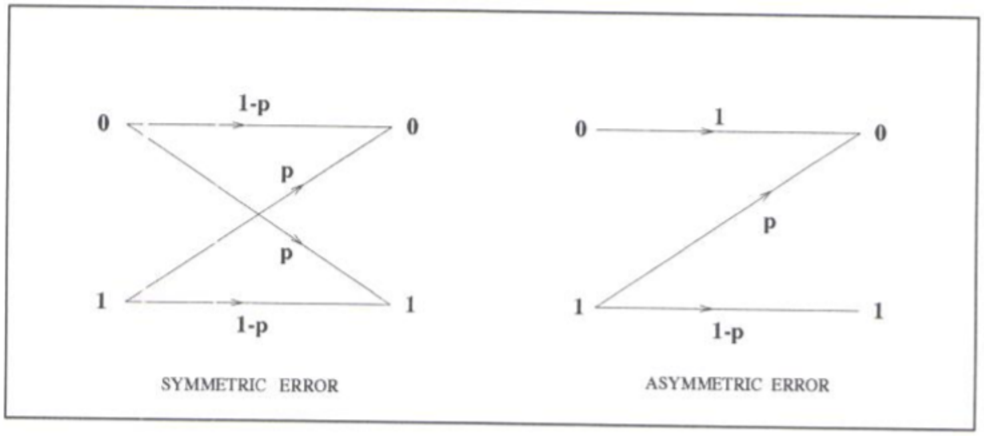

Figure 4.1: Error types

addition errors respectively. However, the Hamming distance measure is not always warranted, because in many environments there is an asymmetry in the type of errors recorded.

In the last two decades, a lot of attention has been paid to the study of codes which are capable of correcting asymmetric or unidirectional errors (Fang 1993). As an example, after analyzing the failures in the cells of semiconductor large scale integrated (LSI) non-volatile memories and metal-nitride-oxide semiconductor (MNOS) memories, Constantin et al. (1979) came to the following conclusion:

"*The LSI and MNOS memories thus exhibit a unidirectional failure property. Although the rest of the memory system is not dependent on power shutoffs and is subject to symmetric failures, for the overall memory systems, the probability of 1 $\rightarrow$ 0 crossover failure is significantly greater than the 0 $\rightarrow$ 1 crossover failure.*"

The asymmetric or unidirectional failure properties of these memories have provided the basis for a new direction of study in coding theory. This research also strongly believes that asymmetry is also exhibited by the human memory system. For example while viewing an object, failure to observe a certain feature present in the object is much more likely than observing a feature that is not present in the object.

Asymmetry is also a natural characteristic of the CyberGlove environment. Consider for example a joint with a 90 range of movement. The resting or reference position of this joint may be anywhere between almost completely extended or almost completely bent. (There seems to be a large variation in this regard among various users.) Suppose that for a particular user the rest position is about 10 from completely extended. Then one may adjust the thresholds to record a "0" for angles less than 10 and record a "1" for angles between 10 and 90.

The angle measurement of any joint tends to vary by a few degrees during different trials of the same sign. There are a number of reasons for this. For example, the user's reference position may change due to health or nervousness. The position of the CyberGlove sensors over the joints may also change slightly. The accuracy of the sensors may also change with the speed of signing. This means that conversion of angles between 15 and 5° is very error prone. In other words, about half of the "0" range but less than seven percent of the "1" range is error prone. Hence addition errors are much more likely. Conversely, if the reference position happens to be closer to 90°, then omission errors will be more likely.

Hence omission errors are much more likely than symmetric errors. For this reason instead of the Hamming distance, this research uses an asymmetric Hamming distance measure (Revesz & Raghava-Rao, 1993). The asymmetric Hamming distance between two feature lists A and B is the maximum of the number of omissions from A or the number of omissions from B that must occur for the two patterns to become indistinguishable. For example, the asymmetric Hamming distance between the signs "there" and "she" in Table 4.1 is two. This is because 00010 can be obtained from 00111 by two omissions and from 01010 by one omission.

Let $\omega(P)$ be the number of 1's in the pattern P. Then we can define the asymmetric Hamming distance more formally as follows.

Definition: *If P_1 and P_2 are two patterns, the asymmetric Hamming distance between P_1 and P_2 is given by*

$A(P_1, P_2) = max(\omega(P_1 \wedge \bar{P}_2, \omega(P_2 \wedge \bar{P}_1))$

In the above definition the complement and the conjunction are bit-wise operations.

4.4 Contour Maps and Reconstructibility

It is easy to see that if at most one omission occurs in the patterns 00111 and 01010, then they remain distinguishable. This is because, if any of the patterns 00110, 00101 or 00011 is received via the CyberGlove, then the system can guess that the user is signing "there", and will not confuse it with "she" which would be registered as either 01010, 01000, or 00010.

Thus the asymmetric Hamming distance can be used as a measure of reconstructibility (after omission errors) of patterns. A pattern is k-reconstructible if it is still distinguishable after k number of omission errors.

Observation: *Considering only omission errors, a set of patterns S is k-reconstructible if and only if for each pair of patterns P_i and P_j in S the Asymmetric Hamming Distance $A(P_i, P_j) \geq k + 1$.*

The reconstructibility of a set of patterns S can be illustrated by a contour map. The contour map shows the reconstructibility regions of each pattern in S. For example, if S is the set of the six feature lists in Figure 4.1, then the contour map will be as in Figure 4.2. There the feature list of each sign is represented within a solid line. The 1-reconstructibility regions of these signs are represented within dashed lines, and the 2-reconstructibility region of "is" is represented within a dotted line.

Every set of patterns can be illustrated similarly. Note that a set of patterns is k-reconstructible if and only if the k-reconstructibility regions do not overlap. In Figure 4.2 the 1-reconstructibility regions do not overlap, hence the feature lists in S are 1-reconstructible.

4.5 Sparse Distributed Memory

The theory sparse distribution memory was introduced by Kanerva (1988). The theory begins with an interpretation of human long-term memory as a storage system that associates sensory input patterns quickly with actions that are appropriate for the situation. In kanerva's model, sensory input is represented in the form of very long bit vectors containing thousands or tens of thousands of bits. Because no two

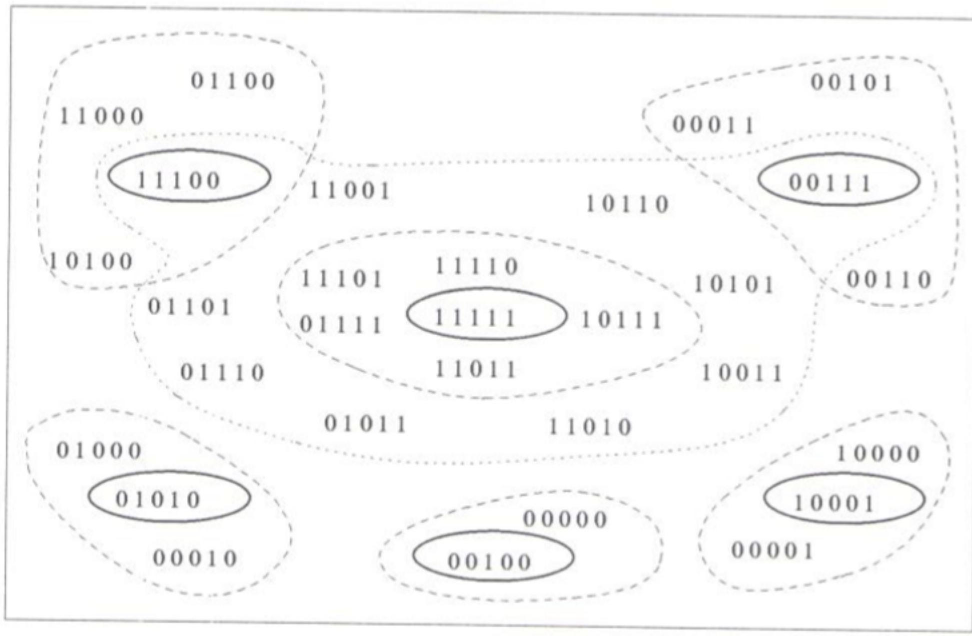

Figure 4.2: Contour Map representation of PIP feature list

external situations are identical, the memory must respond to partial matches between the current sensory pattern and previously stored patterns. Hamming distance is used as the measure of dissimilarity between patterns.

Kanerva proposes an architecture that encompasses an affordable number of physical locations and large pattern size. Each location is assigned an address at random, and the set of location addresses constitutes a sparse subset of the memory space. The memory has an input register for the cue (address) pattern and an input register for the data pattern, and it has a register for an output pattern. Each location has an address decoder that compares its own address with the input cue, selecting that location as a participant in the next storage or retrieval operation whenever the cue is within hamming distance d of the locations address. Kanerva demonstrates that the address decoders can be built of linear threshold circuits.

To store a data pattern at address A, the memory works as follows. The input cue A is presented to the memory, and all locations within a hamming distance of d bits of A select themselves. This set of selected locations is called the sphere selected by A. The copy of the input data pattern, which is to be associated with A, is then entered into each of the selected locations. Because any given location is within the spheres of selection of many distinct cue patterns, entering a new value must not obliterate the previous contents of the location. This is accomplished by implementing each location as a set of counters, one for each bit position of the data. Data are entered by adding 1 to each counter for which the corresponding data bit is 1, and subtracting 1 from each counter for which the corresponding data bit is 0. Kanerva calculates that 8-bit counters are adequate for most applications.

To retrieve a pattern corresponding to input cue A, the memory works as follows. The sphere of selected locations is formed as described above. A set of output counter values is constructed from all the selected locations by summing all the corresponding selected counters; for example, the counter in output position 2 is the sum of the bit-2 counters of each selected physical locations. The output pattern is constructed from the output counters by a threshold method; if an output counter is nonnegative, that output bit is 1; otherwise it is 0.

The sparse distributed memory was used to correct any bit errors that had occurred in the 17-bit CyberGlove feature list. The sparse distribution memory implemented, has a 17-bit register for holding the input which is also used as the address. The memory was designed with 17 counters for each location. Each counter is 8-bit long. Random addresses of 17-bits long are used to simulate the sparse distribution.

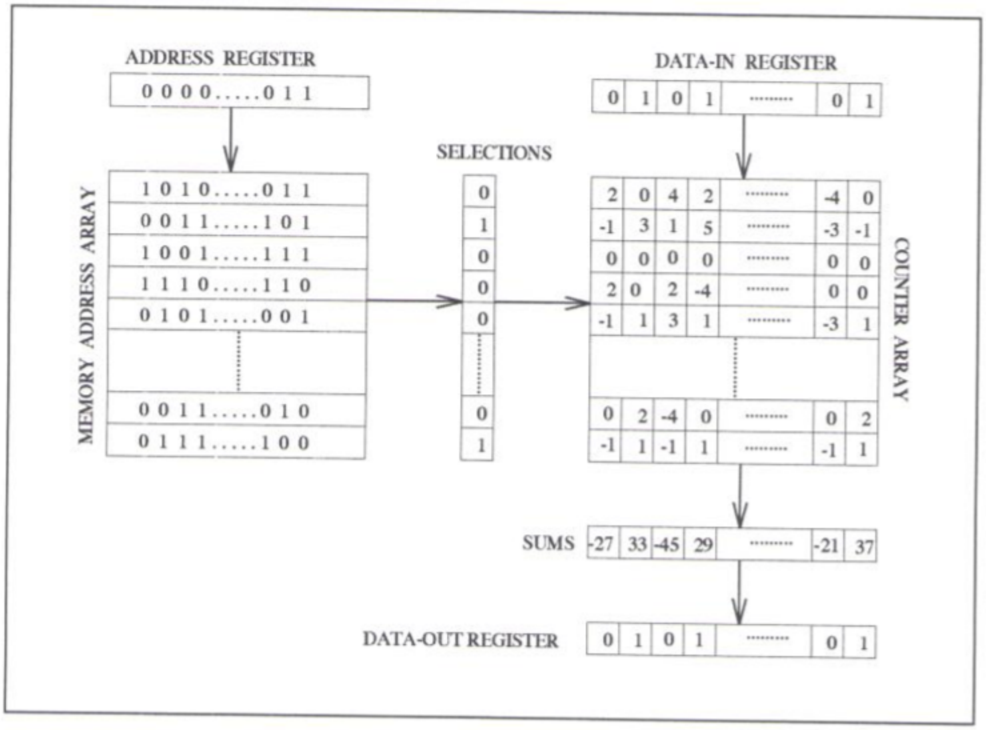

Figure 4.3: Sparse distributed memory

4.6 Matcher Neural Networks

Matcher neural networks (MNNs) are biologically motivated neural networks that were introduced in Revesz (1989,1990). This thesis describes only the algorithmic part of these networks and refers to the original references for their biological context.

Matcher neural networks are especially suitable for the reconstruction of patterns with a large number of omission errors. Like many neural network algorithms, the operation of MNNs can also be divided into two phases: learning and associative recall. The learning part of MNNs is instantaneous, i.e., all the patterns to be learned are memorized instantly and stored in separate memory locations. For each input pattern, the associative recall phase is carried out in four steps.

1. The first step consists of finding those memorized patterns that match the 1's in the input.
2. The second step consists of selecting from the matching patterns those that have the smallest asymmetric Hamming distance from the input.
3. The third step involves finding the percentage of 1's within each group of corresponding bits of the selected patterns.
4. Finally, the fourth step yields as output a binary pattern in which the i^{th} bit is 0 if and only if the percentage of 1's within the i^{th} group is below 50.

As a simple example of the operation of MNNs, consider Figure 4.4. There the first box contains the patterns memorized by the MNN. Note that it is the same set of patterns as shown in Figure 4.1. The pattern to the left of the first box is the input pattern. In this case, the first step of the associative recall phase will find that only the patterns 11111, 00111 and 10001 match all the l's in the input as shown in the second box. The second step will select the pattern 10001 because its asymmetric Hamming distance from the input is only one while the distance of the other two patterns is greater than one. The next two steps of the algorithm in this case are trivial and will result in simply the pattern 10001 to be returned as desired.

Matcher neural networks are similar in many respects to the sparse distributed memories of Kanerva (1988). However, there are several important differences, both

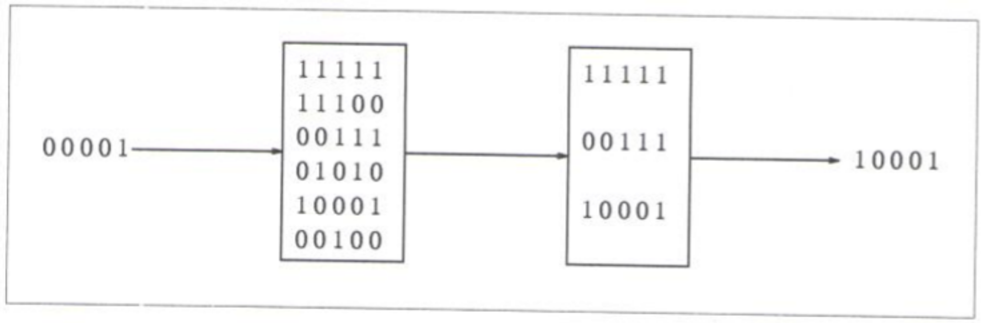

Figure 4.4: Matcher neural network operation

in the biological context and in the algorithmic parts. Four important differences in the algorithmic parts between SDMs and MNNs are that

1. SDMs use distributed storage while MNNs use localized storage for each memorized pattern

2. SDMs use Hamming distances while MNNs use asymmetric Hamming distances

3. SDMs select all patterns whose *addresses* are within a fixed Hamming distance while MNNs select the closest patterns to the input according to the **AHD** measure.

4. SDMs use counters and sum counter values of the selected patterns while MNNs sum corresponding bit values of the selected patterns.

Some of these differences may seem minor, but as we will see in chapter 6 they can significantly effect the relative performance of the two algorithms.

Chapter 5

Automatic Hand Sign Generator

An automatic hand sign generator was designed and implemented with the theory of constraints developed by Revesz (1993). The main idea here is to find a optimal possible set of hand signs that are distinguishable from each other. An algorithm was developed to generate such hand signs. Any two signs so generated will have a different combination of CyberGlove features.

5.1 Feature Analysis

The description of the features extracted from the CyberGlove data and used by the sign-to-speech translation system designed are shown in the figure 5.1. Each hand shape is thus represented by a 17-bit pattern with each bit representing a true or false value associated with a certain feature. So with 17-bits, there are 2^{17} possible patterns or hand shapes. Each hand shape formed can be moved in the six possible directions to generate the conjugations associated with the root word as listed in the Table 5.1. The root word is recognized by the system when the hand is moved to the right followed by a movement to the left. The movements in the other directions generate the conjugations of the root word. For example the movement of the hand to the left followed by a movement to the right generates the plural form of the root

TR	TM	TP	TIA	IM	IP	MM	MP	IMA	RM	RP	MRA	LM	LP	RLA	W_p	W_y

FINGERS
T : THUMB I : INDEX M : MIDDLE R : RING L : LITTLE
JOINTS
M : METACARPOPHALANGEAL JOINT P : PROXIMALINTERPHALANGEAL JOINT
DEGREE OF MOVEMENT
M / P : FLEXION / EXTENSION R : ROTATION A : ABDUCTION Wp : WRIST PITCH Wy : WRIST YAW
BINARY VALUES
0 : EXTENSION / ADDUCTION / UPWARD / NO ROTATION
1 : FLEXION / ABDUCTION / DOWNWARD /ROTATION
NOTE
- FIRST LETTER SIGNIFIES FINGER
- SECOND LETTER INDICATES TYPE OF MOVEMENT OF THE JOINT
- THIRD LETTER INDICATES ABDUCTION / ADDUCTION BETWEEN FINGERS

Figure 5.1: A 17-joint feature list representation of hand signs

word. Assuming that each root word can be conjugated, the system must be capable of translating 6 x 2^{17} gestures to words. Considering the roll of the hand (measured by the flock of birds), the palm of the signer can be oriented upwards or downwards. If it points upward it is decoded as a 0 else it is decoded as a 1. The roll angle of a signer has a wide range of rotation ($\approx 180°$) across the hinge joint. As a result the roll bit is detected correctly 100% of the time. Considering the roll bit the an ideal system must be capable of translating 2 x 6 x 2^{17} gestures to words.

However in English every word does not have a conjugation so the above value serves as an highly optimistic upper bound. Moreover only a small subset of the 2^{17} patterns are signable because of the physical constraints on the hand. On top of this there are errors committed by the signer and also by the CyberGlove while recording. Thus finding an optimal signable set of gestures is a very complex problem to solve. This work attempts to find a good set of such signable gestures.

Table 5.1: Hand movement to word ending

Direction	Feature List	Conjugation
right	1 0 0 0 0 0	root
left	0 1 0 0 0 0	-s
down	0 0 1 0 0 0	-ed
up	0 0 0 1 0 0	-ing
towards	0 0 0 0 1 0	-er
away	0 0 0 0 0 1	-ly -tion -est -meat -en -ness

5.2 Physical Constraints Identification

Using the theory developed by Revesz (1993), this work identifies some of the physical constraints that restrict the number of signable gestures. The structure of the hand was studied in detail in order to solve this problem. Some of the important factors considered are the as follows:

1. **Freedom of movement at each joint**: It is different for different joints. For example PIP joints have one degree of freedom while MCP joints have two degrees of freedom. This parameter is important for feature identification.

2. **Interdigit dependency**: This dependency occurs because of the complex structure of the hand. This is due to the fact that some of the main tendons involved in the movement are split and attached to the bones of several fingers. For example when the index finger MCP and its PIP are flexed the middle finger MCP is also flexed.

3. **CyberGlove sensor measurement**: Some of the sensors in the CyberGlove are more error prone than other. Although the CyberGlove is capable of measuring the arching of the palm due to little finger rotation across the palm,

it is not considered in the present design as it is more error prone. Interdigit dependency also effects some of the measurements made by the CyberGlove.

Following are some of the physical constraints identified using the I7-joint feature list representation of hand signs as shown in Figure 5.1. For example the first physical constraint specifies that when the little finger PIP is flexed and ring finger MCP is extended then it is very difficult to have abduction between the two fingers.

1. (LP = 1) $\wedge$ (RA =0) $\Rightarrow$ (RLA= 0)
2. (MM = I) $\Rightarrow$ (RM = 1)
3. (IM =1) $\Rightarrow$ (IP = 0)
4. (((IM = 1) $\wedge$ (MM = 0)) $\vee$ ((IM = 0) $\wedge$ (MM = 1))) $\Rightarrow$ (IMA = 1)
5. ((MM =0) $\wedge$ (RM =1)) $\Rightarrow$ (MRA=1)
6. (((RM =1) $\wedge$ (LM = 0)) $\vee$ ((RM =0) $\wedge$ (LM = 1))) $\Rightarrow$ (RLA =1)
7. ((IM = 1) $\wedge$ (MM = 1)) $\Rightarrow$ (IMA=0)
8. (MM = 1) $Rightarrow$ (MRA = 0)
9. ((RM = 1) $\wedge$ (LM 1)) $\Rightarrow$ (RLA = 0)
10. (W_p = 1) $\Rightarrow$ (W_y = 0)
11. (RM = 0) $\Rightarrow$ (LM = 0)
12. ((IM = 1) $\wedge$ (IP = 1)) $\Rightarrow$ (MM = 1)
13. ((MP = 1) $\wedge$ (RP = 1)) $\Rightarrow$ (MRA = 1)

14. $((\mathrm{TM} = 1) \land (\mathrm{TP} = 1)) \Rightarrow (\mathrm{TR} = 1)$

15. $(\mathrm{TIA} = 1) \Rightarrow (\mathrm{TR} = 1)$

5.3 Generating 1-Reconstructible Set

The subset A of 2^{17} patterns satisfying the above physical constraints are then used to select the 1-Reconstructible set of patterns. A set of patterns B is 1-Reconstructible if for each pair of patterns in the set the asymmetric hamming distance is greater than or equal to 2. A non-deterministic algorithm was developed to generate the set B of patterns.

1. Choose an empty set B.
2. Choose a pattern randomly from the set A and add it to set B.
3. Delete all those patterns from the set A that are having an asymmetric hamming distance of less than 2 from the randomly selected pattern.
4. Repeat step 2 and step 3 until set A is empty.
5. The set B contains the set of 1-reconstructible signable patterns.

A set of 395 signable patterns were generated by the above algorithm. Note that there is no decision procedure available to prove that the set generated by the above algorithm is optimal. Given n-bits the problem of finding an optimal k-reconstructible set is very difficult combinatorial problem to solve. This research made an unsuccessful attempt to solve this hard problem. Considering the fact that the roll bit is detected correctly 100% of the the time, a set of 395 x 2 = 790 signable patterns can be generated taking the roll bit into account.

5.4 Word Conjugation

In English there are more than 20,000 words. However only a small subset of them can be used as the root words and the rest can be generated from the root words by conjugating them. For example, consider the word "act", it can be conjugated to form words "acts", "acted", "acting", "actor" and "action". Thus each root word selected can be conjugated into six different words in English. Each conjugated word corresponds to the hand displacement in one of the six possible directions.

If we follow the above procedure then we should get 790 x 6 = 4740 conjugated words. However not every word in English can be conjugated into six words. For example, the word "if" cannot be conjugated. On the other hand we cannot throw out all the words that cannot be conjugated as they may be very important in the day to day conversation.

A fair choice of root words is made based on the word-frequency study made by Carroll et al. 1971. Their word frequency study, is essentially an experimental attack on our ignorance of the lexicon. Its purpose is to learn something about the composition and structure of this very large, abstract entity by examining a relatively small, concrete part of it. They tabulated about 86,741 different words along with their frequency of word usage. This research selected first the 395 most frequently used word from Carroll et al. 1971.

These 395 words were doubled to 790 by choosing the antonyms for the originally selected words. For example, if "east" is signed with the palm facing downward then "west" is signed with the palm facing upward. However for words that do not have antonym a closely connected word has been selected. For example, ”cat” and ”rat” are paired. If a word could not be paired then it is replaced from the set with a word that can be paired. Care has been taken to select enough words to represent operations, general objects, pictured objects and qualities in addition to the word-frequency order. Some of the words in Basic English (Ogden 1968) are not chosen as they are no longer frequently used in the modern world.

After selecting the set of 790 root words, each word is conjugated by the suffixes shown in Table 5.1 when the conjugations are meaningful. This process yielded 2718 words. Each of the 395 pairs of words are associated with a 17-joint feature list as shown in Appendix I. Care has been taken to associate the most frequently used words to easily signable gestures to improve the system performance.

Chapter 6

Results and Conclusions

6.1 Testing Methods and Results

The sign to speech translation system was tested using the following approach. Randomly generated hundred signs to be tested and allowed repetition in the signs to be tested following the expected frequency of the signs in ordinary signing. The matcher neural network had in its memory the feature list of each sign to be tested.

The CyberGlove thresholds were calibrated so that the percentage of the angle ranges allocated to 0's was far greater than that allocated to 1's. Each of the hundred CyberGlove generated feature lists was given in sequence as input to the matcher neural network. Noted the number and types of errors in the CyberGlove generated feature lists and in the feature lists reconstructed by the matcher neural network. (Note that the order of the trials is immaterial as far as the statistics are concerned.) For comparison the CyberGlove-generated feature lists were entered into the SDM. The SDM was autoassociative, had hundred randomly generated addresses and 8 bit counters. The SDM was tested with various values of Hamming distance.

The results obtained are listed in the Table 6.1. Out of the hundred input pat-

Table 6.1: Performance of MNN vs. SDM

	CYBERGLOVE OUTPUT	MNN OUTPUT	SDM OUTPUT
Correct	45%	96%	33%
False detection	0%	2%	25%
Misses	55%	2%	42%

terns only 45 were correct, while 41 had a 1-bit error and 14 had 2-bit errors. Out of the total of 69 errors 67 were omission and two were deletion errors. Taking this as input the matcher neural network reconstructed 96 patterns correctly. This 96 is the sum of the 45 that were initially correct plus the 40 cases of with single omission errors and 11 cases with double omission errors. In other words 51 out of 55, that is, 93% of the incorrect inputs were corrected by the matcher neural network.

For the SDM working on the same inputs the value of 4 for the Hamming distance selection was the best. However even in that case only 33% of the outputs were correct, and some of the outputs had up to eleven bit errors. (Using asymmetric Hamming distances with the SDM resulted in even worse performance. It was best with a distance value of 5 yielding only 30 % correctness.) Note that the SDM had a negative effect on the performance of the sign-to-speech translation system because twelve of the correct inputs were actually destroyed. While this difference in the performance of SDMs and MNNs on this problem is not really surprising, it underscores the importance of the differences between these two types of networks and that they are best for different applications.

Table 6.2 lists the error distribution among various CyberGlove features. Thumb rotation and thumb MCP are more error prone than other features. This because of the small range of movement associated with these joints. Note also that omission

Table 6.2: Error distribution among features

FEATURE	OMMISION ERRORS	ADDITION ERRORS
TR	45%	0%
TM	14.49%	0%
LM	5.80%	0%
RM	1.45%	0%
MM	1.45%	0%
IM	4.35%	0%
RLA	1.45%	1.45%
MRA	0%	0%
IMA	8.70%	0%
TIA	11.59%	1.45%
TP	2.90%	0%
IP	2.90%	0%
MP	0%	0%
RP	0%	0%
LP	0%	0%
WP	0%	0%
WY	0%	0%

errors occur 97.1% of the time while addition errors occur about 2.9% only. This illustrates the asymmetric nature of the errors prevalent in the CyberGlove environment. The performance of the matcher neural network is better because of the fact that it is designed to attack those problems where there is asymmetry in the type of error recorded.

Table 6.3 gives a bit error distribution among the gestures recorded. There were about 41% of one bit errors and about 14% of two bit errors recorded by the CyberGlove. Theoretically MNN is capable of correcting all the one bit errors as shown in the Table 6.3. However the statistics also indicate that MNN has also corrected 10 out of the 14 two bit errors. This is a very encouraging result achieved by MNN. However the mechanism of MNN needs to be understood and analyzed properly in order to answer many questions in nature. MNN looks very simple and trivial, but

Table 6.3: Bit error distribution among patterns

ERROR BITS	CYBERGLOVE OUTPUT	MNN OUTPUT	SDM OUTPUT
1	41	0	6
2	14.	0	11
3	0	2	10
4	0	0	13
5	0	0	12
6	0	1	6
7	0	0	3
8	0	1	3
9	0	0	1
10	0	1.45	1
11	0	0	1
12	0	0	0
13	0	0	0
14	0	0	0
15	0	0	0
16	0	0	0
17	0	0	0

the present research believes 1 hat the interaction between feature lists is some what similar to the learning mechanisms in the brain. The internal study of MNN operation is however out of the scope of the present research.

6.2 Conclusions and Future Directions

This work improved the work of Fels & Hinton (1993) by increasing from 66 to 790 the number of artificial root signs that can be robustly translated to speech. A vocabulary size of 2718 words was achieved with conjugations with an accuracy of over 93% in translation. This research compiled a vocabulary of 2718 words which essentially does the work of 20,000 words of English after considering carefully the various linguistic aspects.

There is no training involved and the learning of MIEN was instantaneous. The operation of the system is simple and can be mastered with some practice. The performance of the strobe network is almost 100% and is designed to be reliable. The association of hand signs to feature lists was done with great care to increase the performance of the system. In associating the signs with English words, we chose the signs that are the easiest to make and have the largest mean asymmetric Hamming distances to other patterns to represent the most frequent English words.

However this research opens up many research directions. Given n-bits the problem of finding an optimal k-reconstructible set has not been studied extensively. Surprisingly this is a very difficult combinatorial problem with the even the best solutions at present giving only some lower and upper bounds on the maximal set sizes for most values of k. Bounds on the maximum cardinality of codes correcting up to four asymmetric/unidirectional errors have been tabulated for codes of length <23 by Fang (1993).

The second research direction includes the study of MNN mechanisms and their hardware implementation. The network representation of MNN was already done by Revesz 1989. The ultimate goal is to find how closely it models the human memory as a long term storage device and how it can be used in the neuro-control.

The third research direction involves the comparison of backpropagation with that of MNNs. This research believes that with a different design strategy backpropagation based sign-to-speech translation system's performance can be improved from that of Fels & Hinton (1993). The major bottleneck with Fels & Hinton's design was that they assigned an output node corresponding to each root word. Thus they have 66 output units. So as the vocabulary size increased the network size also blows up. When the network size increases this research believes that the training time of the network also increases. So in future this research will explore the possibility of a better design and compare its performance with that of MNN's. In future this research also plans to build a hybrid system by bringing together the concepts of backpropagation and MNN's.

Backpropagation algorithm (Rumelhart et al. 1986) is well-known and has been applied to various applications. However this research strongly believes that the mechanisms are not understood properly even today. Given a problem there is no procedure that can calculate the number of input, output and hidden nodes required to solve the problem.

Now a stage has been reached where the problem of recognizing ASL has to be undertaken. If possible more number of sensors may be integrated into the system. For example we can integrate one more CyberGlove and one more Bird into the system designed to attack the problem of recognizing ASL. ASL recognition is a very hard problem to solve but in due course we may see deaf people able to communicate in English using computer aid. This research is a step in that direction.

APPENDIX

No	Word	Freq	T R	T M	L M	R M	M M	I M	R L A	M R A	I M A	T I A	T P	I P	M P	R P	L P	W P	W Y
1	a the	124959 373123	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	at of	23975 146001	1	1	1	1	1	1	0	0	0	1	1	0	1	1	1	1	0
3	and or	133899 21283	0	0	1	1	1	0	0	0	1	0	0	0	0	0	0	0	0
4	from to	22799 121347	1	1	1	1	1	0	0	0	1	1	0	1	1	1	1	1	0
5	in out	99108 12252	1	1	1	1	1	1	0	0	0	1	1	0	0	1	1	0	1
6	is was	60852 40934	1	1	1	1	0	0	0	1	0	0	1	0	0	1	1	0	1
7	I you	25932 50957	1	1	1	1	0	0	0	1	0	1	1	1	0	1	1	1	0
8	as for	32208 39322	0	0	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0
9	that this	47443 23301	1	1	0	0	0	0	1	1	1	1	1	0	0	1	0	0	1
10	he she	46249 13653	1	0	1	1	1	1	0	0	0	1	0	0	0	0	0	0	1
11	it them	47284 11997	1	1	0	0	0	0	1	1	0	1	1	1	1	0	0	1	0
12	are were	35454 17031	1	1	1	1	1	0	0	0	1	1	0	0	1	1	1	0	1
13	by with	20189 30455	1	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	1
14	we with	27620 16452	1	0	1	1	1	0	0	0	1	1	0	0	0	0	0	0	0
15	her his	11375 29268	1	0	1	1	1	1	0	0	0	1	0	0	0	0	1	1	0
16	off on	3873 36482	1	1	0	0	0	0	0	1	1	1	1	1	1	1	0	1	1
17	but if	19196 12907	1	1	0	0	0	0	1	0	1	1	0	1	1	1	0	1	0
18	has had	10369 20511	1	1	1	1	0	0	0	1	1	1	1	0	1	0	0	1	0
19	one two	19976 10085	1	1	1	1	1	0	0	0	1	1	1	0	1	0	1	1	0
20	all some	19976 11534	1	1	1	1	1	1	0	0	1	0	0	0	0	0	0	0	1
21	do an	12695 14696	1	1	1	1	1	1	0	0	0	0	1	0	1	1	0	1	0
22	am be	1294 23746	1	1	1	1	1	1	0	0	0	1	1	0	1	0	1	0	0
23	have keep	22331 2509	0	0	1	1	1	1	0	0	0	0	0	0	0	0	1	1	0
24	when where	15886 5611	0	0	1	1	1	1	0	0	0	0	1	0	0	1	0	1	1
25	down up	7206 12776	1	0	1	1	1	1	0	0	0	1	0	0	1	1	0	1	0

No	Word	Freq	TR	TM	LM	RM	MM	IM	RLA	MRA	IMA	TIA	TP	IP	MP	RP	LP	WP	WY
26	now then	7457 12022	1	1	1	1	0	0	0	1	1	1	1	0	0	0	1	0	1
27	here there	4184 15194	1	1	1	1	1	0	0	0	1	1	1	1	0	1	0	0	0
28	may will	6635 12646	0	0	1	1	1	0	0	0	1	0	1	1	1	1	0	0	1
29	equal not	565 18645	0	0	1	1	1	1	0	0	0	0	1	0	1	1	1	1	1
30	other than	10729 7982	0	1	1	1	1	0	0	0	1	0	1	0	0	0	0	0	0
31	each every	14290 3398	0	1	1	1	1	0	0	0	1	0	1	1	1	0	0	1	0
32	how why	13303 4147	0	1	1	1	1	1	0	0	0	0	0	0	1	0	0	1	0
33	can cannot	15247 1279	0	1	1	1	1	1	0	0	0	0	1	0	0	0	1	0	0
34	because so	4207 11543	0	1	1	1	1	1	0	0	0	0	1	0	0	1	1	1	0
35	few many	2685 12158	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
36	about almost	12496 2324	1	0	1	1	1	0	0	0	1	1	0	1	0	1	1	0	1
37	read write	3057 9846	1	0	1	1	1	0	0	0	1	1	0	1	1	0	0	0	1
38	get use	5700 7009	1	1	1	1	1	1	0	0	0	0	0	1	1	1	1	0	0
39	find know	6916 5655	1	0	1	1	1	1	1	0	0	1	1	0	0	1	1	0	1
40	less more	1366 9992	1	1	0	0	0	0	0	0	1	0	1	1	1	1	1	1	0
41	after before	5915 5275	1	1	1	1	0	0	0	1	1	1	0	1	0	1	1	0	1
42	into onto	10620 403	1	1	1	1	1	0	0	0	1	0	1	1	0	0	1	0	1
43	air water	3673 7194	1	1	1	1	1	1	0	0	0	0	0	0	1	1	1	0	1
44	first last	7655 3030	1	1	1	1	1	1	0	0	0	1	0	0	0	1	1	1	0
45	hear see	2154 8518	1	1	1	1	1	1	0	0	0	1	0	0	1	0	0	0	1
46	who while	7576 2837	0	0	1	1	1	0	0	0	1	0	0	1	1	0	0	0	0
47	like opposite	9696 591	0	0	1	1	1	1	0	0	0	0	1	0	1	0	0	0	1
48	very well	5997 4255	0	1	1	1	1	0	0	0	1	0	0	1	0	0	0	0	1
49	every only	3398 6583	1	0	0	0	0	0	1	1	1	1	0	1	1	0	0	0	1
50	no yes	8483 1317	1	0	1	1	0	0	0	1	0	1	0	0	0	0	0	0	0

No	Word	Freq	T R	T M	L M	R M	M M	I M	R L A	M R A	I M A	T I A	T P	I P	M P	R P	L P	W P	W Y
51	also too	4647 5071	1	0	1	1	0	0	0	1	1	1	0	0	0	1	1	0	0
52	another any	4377 5023	1	0	1	1	1	0	0	0	1	1	0	1	0	1	0	1	0
53	hour time	908 8441	0	0	1	1	0	0	0	1	0	0	0	0	0	1	0	1	0
54	new old	5448 3894	0	1	1	1	0	0	0	1	0	0	1	0	0	1	1	0	1
55	letter word	1738 7532	0	1	1	1	0	0	0	1	0	0	1	0	1	0	0	0	1
56	person people	1196 7989	0	0	1	1	1	0	0	0	1	0	1	1	0	0	0	1	0
57	large little	2777 6204	0	1	0	0	0	0	1	1	1	0	1	1	0	1	0	0	1
58	break make	516 8333	0	1	0	0	0	0	1	1	1	0	1	1	1	0	0	1	1
59	different same	3826 5022	0	1	1	1	1	0	0	1	1	0	1	1	0	1	1	0	1
60	must should	4307 3470	0	1	1	1	1	1	0	0	0	0	1	0	0	1	0	0	1
61	road way	1106 6612	0	1	1	1	1	1	0	0	0	0	1	0	0	1	0	0	1
62	left right	2885 4815	1	0	0	0	0	0	1	1	1	1	0	1	0	1	0	1	0
63	give take	3366 4089	1	0	1	1	1	0	0	1	0	1	0	1	0	1	0	0	1
64	day night	5019 2307	1	0	1	1	1	0	0	0	1	1	0	0	1	1	1	1	0
65	hack front	5862 1438	1	1	0	0	0	0	1	1	1	0	1	1	0	1	0	1	0
66	light sound	2376 4667	1	1	1	1	0	0	0	1	0	1	0	1	1	0	1	1	0
67	big small	34761 3555	1	1	1	1	1	1	0	0	0	0	1	0	1	0	0	0	0
68	done through	1566 5442	1	1	1	1	1	1	0	1	0	1	1	0	0	1	0	0	1
69	say show	3916 2734	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	1
70	again often	3892 2616	1	0	0	1	0	0	0	1	0	1	0	1	0	1	1	1	0
71	great important	3855 2588	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0
72	even ever	4225 2036	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0
73	man woman	5486 750	0	1	1	1	0	0	1	1	0	0	0	1	1	0	0	0	0
74	bad good	6601 5343	0	0	1	1	0	0	0	1	1	0	1	0	0	0	0	0	0
75	come go	467 5388	0	0	1	1	1	0	0	0	1	0	0	0	0	1	0	1	0

No	Word	Freq	TR	TM	LM	RM	MM	IM	RLA	MRA	IMA	TIA	TP	IP	MP	RP	LP	WP	WY
76	always never	2657 3115	0	0	1	1	1	0	0	0	1	0	1	0	1	1	1	1	1
77	complete part	1445 4285	0	1	1	1	1	0	0	0	1	0	1	0	1	1	1	0	1
78	above below	2298 3276	1	0	1	1	0	0	0	1	0	1	0	0	0	1	0	1	0
79	rest work	1183 4358	1	0	1	1	0	0	0	1	0	1	0	1	1	0	0	0	0
80	away toward	3814 1690	1	0	1	1	0	0	0	1	0	1	0	1	1	0	1	0	1
81	look view	4933 393	1	0	1	1	1	0	0	0	1	1	0	0	1	0	0	1	0
82	guess think	637 4636	1	0	1	1	1	1	0	0	0	0	0	0	1	1	0	0	1
83	place position	4240 540	1	1	0	0	0	0	0	0	0	1	1	0	1	1	1	1	0
84	land sea	2953 1812	1	1	0	0	0	0	1	1	0	1	1	0	1	0	0	0	1
85	among between	1308 3324	1	1	1	1	0	0	0	1	0	1	1	0	0	1	0	1	0
86	ask tell	900 3715	1	1	1	1	1	0	0	0	1	0	0	1	1	1	0	0	1
87	name street	3766 748	1	1	1	1	1	1	0	0	0	0	0	0	1	0	1	1	0
88	book page	1453 2831	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0
89	far near	2250 1985	0	0	1	1	0	0	1	1	0	0	0	1	0	1	0	0	1
90	attack help	273 3875	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	1
91	begin end	976 2961	0	0	1	1	1	1	0	0	0	0	0	0	0	1	1	0	1
92	across against	1942 1755	0	0	1	1	1	1	0	0	0	0	0	0	1	1	1	1	0
93	let order	2176 1507	1	0	1	1	0	0	0	1	0	1	0	0	1	0	1	1	0
94	earth sky	2690 976	1	0	1	1	0	0	0	1	1	0	0	0	0	0	0	1	0
95	black white	1556 2085	1	0	1	1	0	0	0	1	1	0	0	0	1	0	0	0	1
96	boy girl	2529 1084	1	0	1	1	0	0	0	1	1	0	0	1	0	1	1	0	1
97	kind mean	2262 1256	1	0	1	1	1	0	0	0	1	1	0	0	0	0	1	0	1
98	house office	2705 469	1	1	0	0	0	0	0	1	1	1	1	0	1	0	1	0	1
99	father mother	785 2343	1	1	0	0	0	0	0	1	0	1	1	1	0	1	1	1	0
100	head tail	2487 620	1	1	0	0	0	0	1	1	0	1	0	1	0	1	0	0	1

No	Word	Freq	TR	TM	LM	RM	MM	IM	RLA	MRA	IMA	TIA	TP	IP	MP	RP	LP	WP	WY
101	high low	2237 857	1	1	1	1	0	0	0	1	0	0	0	0	0	0	1	0	0
102	city town	1843 1219	1	1	1	1	0	0	0	1	1	0	0	1	0	1	1	1	0
103	apart together	414 2629	1	1	1	1	1	0	0	0	1	0	1	0	0	1	1	1	0
104	moon sun	1046 1977	1	1	1	1	1	0	0	0	1	1	1	0	0	0	0	1	0
105	dead live	590 2431	0	0	1	1	0	0	0	1	0	0	0	0	0	0	1	0	1
106	event thought	179 2835	0	0	1	1	0	0	0	1	0	0	0	0	1	0	1	1	0
107	necessary need	679 2281	0	0	1	1	0	0	0	1	0	0	1	1	0	0	0	1	0
108	answer question	2002 895	0	0	1	1	0	0	0	1	0	0	1	1	1	0	1	1	0
109	next previous	2727 150	0	0	1	1	0	0	0	1	1	0	1	1	0	1	0	1	0
110	library school	276 2599	0	0	1	1	1	0	0	0	1	0	0	0	1	1	0	0	1
111	country nation	2357 510	0	0	1	1	1	0	0	0	1	0	0	1	0	1	1	0	0
112	class group	1211 1570	0	1	1	1	0	0	0	1	0	0	1	1	0	1	1	1	0
113	minute second	663 2094	0	1	1	1	1	0	0	0	1	0	0	1	0	1	1	1	0
114	close open	1288 1416	0	1	1	1	1	0	0	0	1	0	1	0	1	0	0	0	1
115	cold hot	1469 1233	1	0	0	0	0	0	1	1	0	1	0	1	1	0	0	1	0
116	enough plenty	2363 320	1	0	1	1	1	0	0	0	1	0	0	1	0	1	0	0	1
117	month year	403 2277	1	0	1	1	1	0	0	0	1	0	0	1	1	0	1	1	0
118	picture photo	2500 156	1	1	0	0	0	1	0	1	1	0	1	1	0	1	1	0	1
119	hard soft	1980 669	1	1	0	0	0	0	0	1	1	1	1	0	0	0	1	1	0
120	hand leg	2316 313	1	1	0	0	0	0	1	0	0	0	1	1	1	1	0	1	0
121	blue red	1071 1557	1	1	0	0	0	0	1	0	1	0	1	1	1	0	0	0	1
122	article paper	253 2372	1	1	0	0	0	0	1	1	1	1	0	0	1	0	0	1	0
123	bottom top	858 1741	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	1
124	door window	1748 841	1	1	1	1	0	0	0	1	1	0	1	1	0	1	0	0	0
125	act play	457 2113	1	1	1	1	1	0	0	0	1	0	1	0	0	0	1	0	0

No	Word	Freq	T R	T M	L M	R M	M M	I M	R L A	M R A	I M A	T I A	T P	I P	M P	R P	L P	W P	W Y
126	instrument music	386 2100	1	1	1	1	1	1	0	0	0	0	0	0	0	1	0	1	0
127	plant tree	1051 1421	0	0	0	0	0	0	1	0	0	0	1	0	0	1	0	0	0
128	1564 French	414 886	0	0	0	0	0	0	1	0	1	0	0	0	1	0	0	0	0
129	room wall	1801 642	0	0	0	0	0	0	1	1	1	0	1	1	1	0	0	0	1
130	amount money	701 1694	0	0	1	1	0	0	0	1	0	0	1	1	0	1	1	0	1
131	short tall	1534 848	0	1	0	0	0	0	1	1	0	0	1	0	1	0	0	1	0
132	common special	1174 1192	0	1	1	1	0	0	0	1	0	0	1	1	1	0	0	0	0
133	bird fish	812 1513	0	1	1	1	0	0	0	1	1	0	1	0	1	0	0	1	0
134	run walk	1473 831	1	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0
135	evening morning	543 1736	1	0	1	1	0	0	0	1	0	1	0	1	0	1	1	1	0
136	fire ice	1227 995	1	1	0	0	0	0	0	0	0	1	1	1	0	1	1	0	1
137	consonant vowel	702 1484	1	1	0	0	0	0	0	0	1	0	0	1	1	1	1	0	1
138	start stop	1087 1081	1	1	0	0	0	0	0	1	1	1	0	0	0	1	1	0	1
139	list sort	1781 375	1	1	0	0	0	0	1	1	0	1	1	0	0	1	0	1	0
140	early late	1439 689	1	1	1	1	0	0	0	1	0	0	0	0	0	1	0	0	1
141	change remain	1854 271	1	1	1	1	0	0	0	1	0	1	0	1	0	0	1	0	0
142	cut join	1757 363	1	1	1	1	0	0	0	1	1	1	0	0	0	0	0	0	1
143	quiet talk	967 1133	1	1	1	1	0	0	0	1	1	1	0	0	0	0	1	1	0
144	rhythm song	574 1525	0	0	0	0	0	0	1	1	1	0	1	1	0	1	0	1	0
145	example sample	1939 128	0	0	1	1	1	0	0	0	1	0	0	1	1	0	1	0	1
146	summer winter	1048 1004	0	1	1	1	0	0	0	1	1	0	0	1	0	1	0	0	0
147	ahead behind	639 1376	1	0	0	0	0	0	0	0	1	1	0	0	1	1	1	0	1
148	plane ship	990 1021	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0
149	mountain river	838 1170	1	0	0	0	0	0	1	0	1	1	0	0	0	0	0	1	0
150	check mark	1024 980	1	0	0	0	0	0	1	0	1	1	0	1	0	1	0	0	1

No	Word	Freq	TR	TM	LM	RM	MM	IM	RLA	MRA	IMA	TIA	TP	IP	MP	RP	LP	WP	WY
151	drink eat	347 1616	1	0	1	1	0	0	0	1	0	1	0	0	0	1	1	0	1
152	add subtract	1654 292	1	1	1	1	1	0	0	1	0	0	1	1	0	0	1	1	1
153	move station	1592 341	1	1	1	1	0	0	0	1	0	0	1	1	1	0	0	1	0
154	learn teach	1674 254	0	0	0	0	0	0	0	1	1	0	1	1	0	1	1	0	1
155	chair table	421 1502	0	0	0	0	0	0	1	0	1	0	1	1	0	0	0	0	0
156	false true	225 1696	0	0	1	1	1	0	0	0	1	0	0	0	1	0	1	1	0
157	bright dark	741 1171	0	0	1	1	1	0	0	0	1	0	0	1	1	1	0	1	0
158	circle square	945 965	0	1	0	0	1	0	1	0	0	0	0	0	0	0	0	0	1
159	car cart	1752 136	0	1	0	0	1	0	1	0	1	0	1	0	1	1	0	1	0
160	rain snow	938 948	0	1	1	1	0	0	0	1	0	0	1	0	0	1	0	1	0
161	avoid face	246 1629	0	1	1	1	0	0	0	1	1	0	0	0	0	0	0	0	1
162	ready wait	1207 656	0	1	1	1	0	0	0	1	1	0	0	0	0	1	1	1	0
163	north south	926 928	1	1	1	1	1	0	0	0	1	0	0	0	1	1	1	0	0
164	seat table	339 1502	1	0	0	0	0	0	0	1	0	0	0	0	0	1	0	1	0
165	animal human	1122 710	1	1	0	0	0	0	1	0	1	1	1	0	0	0	0	1	0
166	call visit	1374 443	1	1	1	1	0	0	0	1	0	0	0	0	1	0	0	1	0
167	measure scale	1056 737	1	1	1	1	0	0	0	1	0	0	0	1	0	1	0	1	0
168	hundred thousand	1187 597	1	1	1	1	1	0	0	0	1	0	1	0	0	1	0	0	1
169	blow wind	411 1336	0	0	0	0	0	0	0	1	0	0	1	0	1	0	1	0	0
170	grow kill	1418 326	0	0	0	0	0	0	0	1	1	0	1	1	1	0	1	1	0
171	flat round	662 1076	0	0	1	1	0	0	0	1	0	0	0	1	0	1	1	1	0
172	base unit	881 838	0	0	1	1	0	0	0	1	1	0	1	0	0	0	1	1	0
173	quite rather	967 734	0	0	1	1	0	1	0	1	1	0	1	1	0	0	0	0	1
174	size weight	1057 610	0	0	1	1	1	0	0	0	1	0	1	0	0	1	1	0	1
175	language sign	1041 615	0	1	1	0	0	0	0	1	0	0	0	0	1	0	0	0	0

No	Word	Freq	TR	TM	LM	RM	MM	IM	RLA	MRA	IMA	TIA	TP	IP	MP	RP	LP	WP	WY
176	sit stand	549 1081	0	1	0	0	0	0	1	0	0	0	1	1	1	1	0	0	1
177	fall spring	824 802	0	1	1	1	0	0	0	1	1	0	1	0	0	1	0	0	1
178	dance sing	608 1014	1	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	1
179	beside else	725 859	1	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
180	cool warm	500 1072	1	1	0	0	0	0	0	1	0	1	0	0	0	0	1	0	0
181	fast slow	1173 396	1	1	0	0	0	0	0	1	0	0	1	1	1	0	1	1	0
182	correct wrong	940 606	1	1	0	0	0	0	0	1	0	1	0	1	1	0	1	0	1
183	follow lead	1022 520	1	1	1	1	1	0	0	0	1	0	0	0	1	1	0	1	0
184	empty full	384 1144	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0
185	cow horse	263 1263	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	1
186	edge middle	817 706	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	1	0
187	dog pig	1380 142	0	0	0	0	0	0	0	0	1	0	1	1	1	1	1	0	0
188	farm garden	900 600	0	0	0	0	0	1	) 0	1	0	0	1	0	0	0	0	1	0
189	difficult easy	592 894	0	0	0	0	0	0	1	1	0	0	1	0	1	0	0	0	1
190	America Britain	1321 149	0	0	0	0	0	0	1	1	0	0	1	1	0	1	0	0	1
191	future past	354 1109	0	1	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0
192	either neither	1033 402	0	1	0	0	0	0	0	0	1	0	1	1	0	1	1	1	0
193	poor rich	851 584	0	1	0	0	0	1	1	0	0	0	1	1	0	0	0	0	1
194	certain doubt	1198 222	0	1	0	0	0	0	1	1	0	0	0	1	1	0	0	0	1
195	dry wet	993 418	0	1	0	1	1	0	1	1	1	0	1	1	0	1	0	0	0
196	gold silver	895 502	1	0	0	0	0	0	0	0	0	1	0	0	0	1	0	1	0
197	strong weak	1140 245	1	0	0	0	0	0	0	0	1	1	0	1	1	0	1	1	0
198	iron steel	817 565	1	0	0	0	0	0	0	1	1	1	0	1	0	0	1	0	1
199	able stable	1260 114	1	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	1
200	brain heart	330 1032	1	0	0	0	0	0	1	0	1	1	0	0	1	1	0	0	0

No	Word	Freq	T R	T M	L M	R M	M M	I M	R L A	M R A	I M A	T I A	T P	I P	M P	R P	L P	W P	W Y
201	buy sell	872 462	1	0	0	0	0	0	1	1	0	1	0	0	0	1	0	1	0
202	history science	726 602	1	0	1	1	0	0	0	1	1	0	0	1	0	0	1	1	0
203	peace war	343 968	1	0	1	1	1	0	0	0	1	0	0	0	1	1	1	0	1
204	clock watch	330 969	1	1	0	0	0	0	0	0	1	1	1	1	0	1	0	1	0
205	copy original	887 403	1	1	0	0	0	0	1	0	0	0	1	1	0	0	0	1	0
206	narrow wide	426 863	1	1	0	0	0	0	1	0	0	1	1	0	1	1	0	0	0
207	building property	857 419	1	1	1	1	0	0	0	1	0	0	1	0	0	0	1	1	0
208	fact observe	925 349	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0	0
209	sand stone	672 593	0	0	0	0	0	0	1	0	0	0	1	0	1	0	0	1	0
210	bottle glass	346 913	0	0	0	0	0	0	1	0	0	0	1	1	1	1	0	1	0
211	ancient modern	515 731	0	0	0	0	0	0	1	1	0	0	0	0	0	1	0	0	1
212	bring remove	1016 215	0	0	0	0	0	0	1	1	0	0	0	1	0	0	0	1	0
213	friend enemy	923 301	0	0	0	0	0	0	1	1	1	0	0	1	0	1	0	0	0
214	clear noise	811 411	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	1	0
215	ball net	1061 157	0	1	0	0	0	0	1	0	1	0	0	0	0	0	0	1	0
216	floor roof	935 276	1	0	0	0	0	0	0	0	0	1	0	0	1	0	1	1	0
217	memory mind	139 1046	1	0	0	0	0	0	0	0	0	1	0	1	1	0	0	0	0
218	beautiful ugly	1048 126	1	0	0	0	0	0	0	1	0	1	0	0	1	0	1	0	1
219	control rule	556 618	1	0	0	0	0	0	0	1	0	1	0	1	0	0	0	1	0
220	complex simple	206 956	1	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0
221	impossible possible	231 930	1	0	0	0	0	0	0	1	1	1	0	0	0	0	1	1	0
222	thick thin	540 611	1	0	0	0	0	0	1	0	1	0	0	1	1	1	0	1	0
223	coal oil	419 731	1	1	1	0	0	0	1	0	0	1	1	1	0	0	0	1	1
224	border center	155 984	1	1	0	0	0	0	1	1	1	0	0	0	0	1	0	1	0
225	chemical natural	376 739	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	1

No	Word	Freq	TR	TM	LM	RM	MM	IM	RLA	MRA	IMA	TIA	TP	IP	MP	RP	LP	WP	WY
226	skin touch	2237 857	0	0	0	0	0	0	0	0	1	0	1	0	1	0	0	0	1
227	business market	1843 1219	0	0	0	0	0	0	0	1	1	0	1	0	0	0	1	1	0
228	happy sad	414 2629	0	0	0	0	0	0	1	1	0	0	1	0	0	1	0	1	0
229	fly swim	1046 1977	0	1	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0
230	journey travel	590 2431	0	1	0	0	0	0	0	1	0	0	1	0	0	1	1	0	1
231	salt sugar	179 2835	0	1	1	1	0	0	0	1	0	0	1	0	0	0	1	0	1
232	purpose reason	679 2281	0	1	1	1	1	0	0	0	1	0	1	0	0	1	1	1	0
233	wood wax	2002 895	1	0	0	0	0	0	0	0	1	1	0	0	0	1	1	0	0
234	daughter son	2727 150	1	0	0	0	0	0	0	1	1	1	0	0	0	1	0	0	1
235	cheif general	276 2599	1	0	0	0	0	0	1	0	0	1	0	1	1	0	0	0	1
236	ear eye	2357 510	0	0	1	1	1	1	0	0	0	0	0	0	1	1	0	0	0
237	east west	1211 1570	1	0	1	1	1	0	0	0	1	0	0	0	1	1	1	1	0
238	industry produce	663 2094	1	1	0	0	0	0	0	0	0	0	1	0	1	1	1	0	1
239	mouth tongue	1288 1416	1	1	0	0	0	0	0	0	0	1	1	1	0	0	1	1	0
240	fat thin	1469 1233	1	1	0	0	0	0	0	1	0	1	0	0	1	0	1	1	0
241	brother sister	2363 320	1	1	0	0	0	0	0	1	0	1	1	0	1	1	0	0	0
242	column row	403 2277	1	1	0	0	0	0	0	1	1	0	1	0	0	0	1	0	1
243	bend straighten	2500 156	1	1	0	0	0	0	0	1	1	1	1	1	0	0	0	0	1
244	metal powder	1980 669	1	1	0	0	0	0	1	0	0	1	0	0	1	0	0	0	0
245	train truck	2316 313	1	1	1	0	1	0	1	1	0	0	0	0	0	0	0	0	0
246	fix free	1071 1557	1	1	0	0	0	0	1	1	0	0	1	0	0	1	0	0	1
247	catch throw	253 2372	1	1	0	0	0	0	1	1	1	0	1	0	1	0	0	0	0
248	comb hair	858 1741	0	0	1	0	0	0	0	0	0	0	1	0	0	1	1	0	1
249	nose smell	1748 841	0	0	0	0	0	0	0	0	1	0	1	1	0	0	1	0	1
250	temperature thermometer	457 2113	0	0	0	0	0	0	0	1	0	0	1	1	0	1	0	0	0

No	Word	Freq	T R	T M	L M	R M	M M	I M	R L A	M R A	I M A	T I A	T P	I P	M P	R P	L P	W P	W Y
251	accompany alone	90 825	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	1
252	material substance	651 260	0	0	0	0	0	0	0	1	1	0	0	1	0	0	1	0	0
253	arm army	491 412	0	0	0	0	0	0	1	1	1	0	0	1	1	0	0	1	1
254	meal meat	286 617	0	0	1	1	0	0	0	1	1	0	0	0	0	1	1	0	1
255	cause effect	502 399	0	0	1	1	0	0	0	1	1	0	0	1	1	0	0	1	0
256	pull push	558 339	0	1	0	0	0	1	0	1	1	0	0	1	0	0	1	1	0
257	king queen	688 206	0	1	0	0	0	0	1	0	0	0	0	1	0	1	0	0	0
258	normal strange	179 710	1	0	0	0	0	0	0	0	1	0	0	0	1	1	0	0	0
259	husband wife	233 636	0	0	1	1	0	0	0	1	1	0	1	1	0	1	0	1	0
260	adult child	132 730	0	0	1	1	1	0	0	0	1	0	0	0	1	1	0	0	1
261	market trade	383 479	1	0	0	0	0	0	1	1	0	0	0	0	1	0	0	1	0
262	note slip	713 149	1	1	0	0	0	0	0	0	1	1	1	0	1	0	1	1	0
263	drive sail	543 312	1	1	0	0	0	0	0	1	0	0	1	0	0	1	1	1	0
264	awake sleep	134 717	1	1	0	0	0	0	0	1	1	0	0	1	0	1	1	1	0
265	hate love	107 735	1	1	0	0	0	0	1	0	1	0	1	0	0	0	0	0	0
266	danger safe	359 474	1	1	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0
267	liquid solid	442 390	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1	1	0
268	balance interest	263 560	0	0	0	1	0	0	1	0	1	0	0	1	1	1	0	0	1
269	major minor	597 223	0	1	0	0	0	0	0	0	1	0	0	0	0	0	1	0	1
270	save spend	396 418	0	1	0	0	0	0	0	0	1	0	0	0	1	1	1	1	0
271	billion million	147 662	0	1	0	0	0	0	0	1	1	0	0	0	1	0	1	0	1
272	divide multiply	465 337	0	1	0	0	0	0	1	0	0	0	0	0	1	1	0	0	1
273	branch root	233 568	0	1	0	0	0	0	1	0	0	0	0	1	1	0	0	1	0
274	dinner lunch	430 357	0	1	0	0	0	0	1	1	1	0	0	0	0	0	0	0	1
275	key lock	652 123	1	0	0	0	0	0	0	0	0	1	0	0	1	1	0	0	1

No	Word	Freq	TR	TM	LM	RM	MM	IM	RLA	MRA	IMA	TIA	TP	IP	MP	RP	LP	WP	WY
276	absent present	36 732	1	0	0	0	0	0	0	0	1	1	0	1	0	0	0	1	1
277	rod stick	259 502	1	0	0	0	0	0	0	1	0	0	0	1	0	1	1	1	0
278	bread cake	515 244	1	1	0	0	0	0	0	0	0	1	1	0	0	0	1	1	1
279	crime trouble	52 695	1	1	0	0	0	0	0	1	0	0	1	0	0	0	0	0	1
280	gun knife	422 318	1	1	0	0	0	0	0	1	0	0	1	1	1	0	0	0	1
281	cotton silk	554 184	1	1	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0
282	receive send	246 485	1	1	0	0	0	0	1	0	1	0	0	0	1	1	0	0	1
283	decide judge	540 173	0	0	0	0	0	0	0	1	0	0	1	1	0	1	1	1	0
284	backward forward	121 591	0	0	0	0	0	0	0	1	0	0	1	1	1	0	1	0	1
285	range rate	351 359	0	0	0	0	0	0	0	1	1	0	0	0	1	0	0	1	0
286	process theory	468 240	0	1	0	0	0	0	0	1	0	0	0	1	0	0	0	0	1
287	cat rat	620 85	0	1	0	0	0	0	0	1	1	0	0	0	0	1	0	1	0
288	drop lift	433 267	0	1	0	0	0	0	1	0	1	0	0	1	0	1	0	1	0
289	attempt chance	138 548	1	0	0	0	0	0	0	0	0	1	0	1	0	1	1	0	1
290	desire hope	129 544	1	0	0	0	0	0	0	0	0	1	0	1	1	1	0	1	0
291	birth death	147 518	1	0	0	0	0	0	0	0	1	0	0	1	0	1	1	1	0
292	clean dirty	521 143	1	0	0	0	0	0	1	0	1	1	0	0	0	1	0	0	1
293	gentlemen lady	121 539	1	0	0	0	0	0	1	1	1	1	0	0	1	0	0	0	1
294	marry ring	141 516	1	1	0	0	0	0	0	0	0	0	1	0	0	0	1	1	0
295	courage fear	219 437	1	1	0	0	0	0	0	0	1	0	1	0	1	0	1	0	0
296	angle degree	462 191	1	1	0	0	0	0	0	0	1	1	0	0	1	1	0	1	0
297	dull sharp	147 499	1	1	1	0	0	0	0	0	1	1	0	1	0	1	1	0	0
298	angle degree	43 590	1	1	0	0	0	0	0	0	1	1	1	0	0	1	0	0	1
299	prefix suffix	200 428	1	1	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0
300	sudden surprise	235 389	1	1	0	0	0	0	0	1	0	0	1	1	0	0	1	0	0

No	Word	Freq	T R	T M	L M	R M	M M	I M	R L A	M R A	I M A	T I A	T P	I P	M P	R P	L P	W P	W Y
301	rough smooth	292 331	1	1	0	0	0	0	1	0	1	0	0	1	0	0	0	0	1
302	describe detail	486 136	1	1	0	0	0	0	1	0	1	1	0	1	1	0	0	1	0
303	tomorrow yesterday	362 257	0	0	0	0	0	0	0	0	0	0	1	0	1	1	1	1	0
304	combine separate	200 416	0	0	0	0	1	0	0	1	0	0	0	1	0	1	1	0	1
305	cry laugh	327 287	0	0	0	0	0	0	0	1	1	0	1	0	0	1	0	0	1
306	private public	163 447	0	1	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0
307	grain rice	340 264	0	1	0	0	0	0	0	0	0	0	1	0	0	1	0	1	0
308	bag basket	371 227	0	1	0	0	0	0	0	0	1	0	1	0	1	1	0	0	1
309	goat sheep	130 464	0	1	0	0	0	0	0	1	0	0	0	1	1	0	1	0	1
310	cent dollar	364 217	1	0	0	0	0	0	0	1	1	1	0	1	1	0	0	0	0
311	neck throat	373 207	1	0	0	0	0	0	1	0	0	0	0	1	0	1	0	1	0
312	cheap valuable	60 512	1	0	0	0	0	0	1	0	1	0	0	1	1	0	0	0	0
313	desk shelf	421 150	1	1	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0
314	paint print	437 134	1	1	0	0	0	0	0	0	0	1	0	1	1	1	0	0	1
315	fruit vegetable	456 112	1	1	0	0	0	0	0	1	1	1	0	0	1	0	0	0	1
316	military political	209 356	1	1	0	0	0	0	1	0	0	1	0	0	0	1	0	0	1
317	glove hat	42 511	1	1	0	0	0	0	1	0	0	1	0	1	0	1	0	1	0
318	meter mile	206 344	0	0	0	0	0	0	0	0	1	0	0	0	1	1	1	1	0
319	fog smoke	212 328	0	0	0	0	0	0	1	0	1	0	0	0	1	1	0	1	0
320	copper tin	326 203	0	1	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0
321	loud mild	410 115	0	1	0	0	0	0	0	0	1	0	1	0	0	1	1	0	0
322	pen pencil	194 331	0	1	0	0	0	0	0	1	1	0	1	0	0	0	0	0	1
323	price tax	289 233	1	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0
324	finger thumb	370 150	1	0	0	0	0	0	0	1	0	0	0	1	1	0	1	1	0
325	apple orange	294 221	1	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1

No	Word	Freq	TR	TM	LM	RM	MM	IM	RLA	MRA	IMA	TIA	TP	IP	MP	RP	LP	WP	WY
326	coat skirt	391 123	1	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	1
327	butter cheese	275 236	1	1	0	0	0	0	0	0	0	1	0	0	0	1	1	1	0
328	brush wash	251 252	1	1	0	0	0	0	0	0	0	1	0	1	0	1	0	0	0
329	coffee tea	280 223	1	1	0	0	0	1	0	0	1	0	0	1	0	0	0	1	0
330	company organization	339 161	1	1	0	0	0	0	0	0	1	0	1	1	1	0	0	1	0
331	height width	346 149	1	1	0	0	0	0	0	1	0	0	0	1	0	1	1	0	1
332	leather wool	226 255	1	1	0	0	0	0	0	1	1	0	1	1	0	1	0	0	0
333	healthy sick	140 334	1	1	0	0	0	0	1	0	0	0	0	0	1	1	0	1	0
334	discover invent	364 73	1	1	0	0	0	0	1	0	0	0	1	0	1	0	0	0	1
335	loose tight	227 208	0	0	0	0	0	0	0	0	1	0	1	1	0	1	0	1	0
336	jump stamp	356 76	0	0	0	0	0	0	0	1	0	0	1	1	1	0	0	1	0
337	bar hotel	325 106	0	1	0	0	0	0	0	0	0	0	1	1	1	0	1	1	0
338	opinion suggest	182 244	0	1	0	0	0	0	0	1	1	0	0	1	0	1	0	0	1
339	bitter sweet	103 319	0	1	0	0	0	0	0	1	1	0	1	1	1	0	0	0	0
340	boil steam	81 340	1	0	0	0	0	0	0	0	1	0	0	1	1	0	1	0	1
341	data date	157 261	1	0	0	0	0	0	0	1	1	0	0	0	0	0	1	0	1
342	negative positive	210 204	1	1	0	0	0	0	0	0	0	0	0	1	0	0	1	0	1
343	roll slope	266 146	1	1	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0
344	knot thread	82 317	1	1	0	0	0	0	0	0	0	1	0	1	1	0	1	0	0
345	medical physical	149 240	1	1	0	0	0	0	0	1	1	1	0	1	0	0	0	1	0
346	atom molecule	225 160	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	0	1
347	protest support	43 341	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	1
348	miss waste	191 191	0	0	0	0	0	0	0	0	1	0	0	1	0	1	0	0	1
349	bus van	345 35	0	1	0	0	0	0	0	0	0	0	0	1	1	1	1	0	1
350	foolish wise	126 253	0	1	0	0	0	0	0	0	0	0	1	0	1	0	1	0	1

No	Word	Freq	T R	T M	L M	R M	M M	I M	R L A	M R A	I M A	T I A	T P	I P	M P	R P	L P	W P	W Y
351	pain pleasure	198 166	1	1	0	0	0	0	0	1	0	0	0	1	0	1	0	1	0
352	religion respect	179 185	1	1	0	0	1	0	0	1	1	0	0	0	1	1	0	1	0
353	January July	156 199	1	1	0	0	0	0	0	1	1	0	0	1	1	0	1	0	0
354	leaf stem	214 138	0	1	0	0	0	0	0	1	0	1	0	0	0	0	1	1	0
355	pipe pump	232 117	0	0	0	0	0	0	0	0	1	0	0	1	1	0	1	1	0
356	offer request	185 159	0	1	0	0	0	0	0	0	0	0	0	1	0	1	1	1	0
357	burn flame	190 151	1	1	0	0	0	0	0	0	1	0	0	1	1	1	0	0	0
358	quality quantity	259 74	0	1	0	0	0	0	0	1	1	0	1	0	1	0	0	1	0
359	smile weep	281 36	1	1	0	0	0	0	0	0	0	0	1	1	0	1	0	0	1
360	hospital wound	154 161	1	1	1	0	0	0	0	0	1	0	0	0	0	1	1	0	1
361	oven store	128 183	1	1	0	0	0	0	0	0	1	0	0	0	1	0	0	0	1
362	hammer nail	155 153	1	1	0	0	0	0	0	0	1	0	1	0	0	1	0	1	0
363	poison snake	76 226	1	1	0	0	0	0	0	1	0	1	0	1	0	0	0	0	1
364	assume condition	99 193	0	0	0	0	0	0	0	0	1	0	0	0	1	1	1	0	1
365	soup wine	184 108	0	0	0	0	0	0	0	1	1	0	0	1	1	0	0	0	1
366	female male	147 139	1	1	1	1	0	0	0	1	1	1	1	1	0	1	0	0	1
367	shirt trouser	222 60	1	1	1	1	0	0	0	1	1	0	1	1	1	0	1	1	0
368	insect worm	202 73	1	1	1	1	1	0	0	0	1	0	1	1	1	1	0	1	0
369	ant bee	146 128	1	1	1	1	0	0	0	1	0	1	1	0	0	0	0	0	0
370	float sink	125 149	0	1	1	1	1	0	0	0	1	0	0	1	0	1	1	0	1
371	decrease increase	29 244	0	1	1	1	1	1	0	0	0	0	0	0	1	0	1	1	1
372	hang hook	159 112	1	1	0	0	0	0	0	1	1	1	0	1	1	0	1	1	0
373	muscle nerve	133 121	0	1	1	1	0	0	0	1	0	0	1	1	1	0	1	0	1
374	shade umbrella	203 51	1	1	1	1	1	0	0	0	1	0	0	0	0	0	0	1	1
375	fold twist	168 81	1	1	1	1	1	0	0	0	1	1	1	0	0	1	0	0	0

No	Word	Freq	TR	TM	LM	RM	MM	IM	RLA	MRA	IMA	TIA	TP	IP	MP	RP	LP	WP	WY
376	bath soap	95 134	1	1	1	1	1	0	0	0	1	1	0	0	1	0	1	0	0
377	loss profit	162 67	0	1	0	0	0	0	0	0	1	0	1	1	1	0	1	0	1
378	anger content	108 114	1	1	1	1	0	0	0	1	0	0	0	0	1	0	1	0	1
379	shoe socks	146 73	0	0	1	1	0	0	0	1	0	0	0	1	1	0	1	0	1
380	accept refuse	159 57	0	0	1	1	0	0	0	1	1	0	0	1	0	0	1	0	0
381	burst smash	194 20	1	1	0	0	0	0	0	1	1	1	1	1	1	0	1	0	1
382	collect distribute	189 20	1	1	1	1	0	0	0	1	0	1	1	1	1	0	1	0	1
383	digest stomach	23 184	1	1	1	1	0	0	0	1	0	1	1	0	1	0	0	0	1
384	delicate violent	114 80	1	1	1	1	1	0	0	0	1	1	1	0	1	0	0	0	1
385	contract expand	94 87	1	0	1	1	0	0	0	1	1	1	0	1	1	0	0	1	0
386	crack crush	144 23	1	1	1	1	1	0	0	0	1	1	0	1	0	0	1	1	0
387	nut screw	59 100	1	1	1	1	1	1	0	0	0	0	1	0	0	0	1	0	1
388	credit debt	100 43	1	1	1	1	0	0	0	1	1	0	0	1	1	0	1	0	1
389	manager secretary	85 55	0	0	1	1	0	0	0	1	1	0	1	0	1	0	1	0	1
390	kick ass	89 43	1	1	1	1	1	0	0	1	1	0	1	1	1	0	1	0	0
391	attract repel	101 22	1	1	1	1	0	0	0	1	1	0	0	1	1	0	0	0	0
392	punish reward	29 84	0	1	1	1	0	0	0	1	1	0	0	1	1	0	1	1	0
393	dense sparse	86 14	0	1	1	1	0	0	0	1	1	0	0	0	1	0	1	0	0
394	approval regret	51 40	0	1	1	1	0	0	0	1	1	0	0	1	0	0	1	0	1
395	couch mat	47 37	1	1	1	1	1	0	0	0	1	0	1	1	1	1	1	0	1

Bibliography

[1] April, E.W., Anatomy, National Medical Series, Harwal Publishing Company, 1990.

[2] Ascension Technologies inc., The Flock of BirdsTM installation and Operation guide, 1992.

[3] Bose, B., Rao, T.R.N; Theory of unidirectional error correcting/detecting codes, iEEE Trans. Comput., C-31, 521-530, June 1982.

[4] Brouwer, A.E, Shearer, J.B, Sloane, N.J.A, Smith, W.D, A new table of constant weight codes, IEEE Transactions on information Theory, Vol 36 01), 1990.

[5] Carroll, J.B, Davies, P. Richman, B. Word Frequency Book Houghton Mifflin Company, 1971.

[6] Constantin, S.D., Rao, T.R.N., On the theory of binary asymmetric error correcting codes, information and Contr., 40, 20-36, 1979.

[7] Digital Equipment Co., $DECtalk^{TM}$ Reference manual, 1992.

[8] Fang Gang, Binary Block Codes for Correcting Asymmetric or Unidirectional Errors, Ph D Dissertation, Eindhoven University of Technology, Netherlands, 1993.

[9] Fels,S.S., Building Adaptive interfaces with Neural Networks: the Glove-Talk

Pilot Study, Technical Report CRG-TR-90-1, University of Toronto, February 1990.

[10] Fels, S.S., Hinton, G.E., Glove-Talk: A neural network interface between a Data-Glove and a speech synthesizer, IEEE Transactions on Neural Networks, 4 (1), 2-8, 1993.

[11] Guillory, L. Expressive and receptive fingerspelling for hearing adults, Baton Ruge, 1966.

[12] Jones, L.V., Wepman, J.M. A Spoken Word Count, Language Research Associates, 1966.

[13] Kanerva, P., Sparse Distributed Memory, MIT Press, 1988.

[14] Kramer, J., Leifer, L., "The Talking Glove: An expression and receptive "Verbal" Communication Aid for the Deaf, Deaf-Blind, and Nonvocal," in Proceedings of the third annual conference "Computer Technology/Special Education/Rehabilitation," California State University, Northridge, 335-340, 1987.

[15] Kramer, J., Leifer, L. The 'TalkingGlove': A speaking aid for nonvocal deaf and deaf-blind individuals, Proc. of 12th Annual RESNA Conference, 47i-472, Louisiana, USA, 1989.

[16] Loomis, J., Poizner, H., Bellugi, U., Blakemore, A., Hollerbach, J., Computer graphic modeling of American Sign Language, Computer Graphics, 17 (3), 105114, 1983.

[17] Mandel, M., "Phonotactics and Morphophonology in American Sign Language," Doctoral Dissertation, University of California, Berkeley, 1981.

[18] Mayberry, R., Manual Communication, in H. Davis and S.R.Silverman(Eds), Hearing and Deafness, New York.

[19] MacWilliams, F.J., Sloane, N.J.A, The Theory of Error Correcting Codes, 1981.

[20] Meier, R.P., Language acquisition by deaf children, American Scientist, 79, 60-70, 1991.

[21] Ogden, C.K., Basic English: international Second Language, World inc, NY, USA, 1968.

[22] Poizner, H., Perception of movement in American Sign Language: Effects of linguistic structure and linguistic experience, Perception 61 Psychophysics, 33 (3), 215-231, 1983.

[23] Poizner, H., Klima, E.S., Bellugi, U., What the hands reveal about the brain, The MIT Press, 1987.

[24] Revesz, P.Z., Matcher neural networks, lrst international Joint Conference on Neural Networks, vol. 1, 767-772, Washington D.C., 1989.

[25] Revesz, P.Z., Functional interpretations of neocortical modules, 2nd international Joint Conference on Neural Networks, vol. 2, 509-514, San Diego, CA, 1990.

[26] Revesz, P.Z., Raghava-Rao K. V., A Sign-to-Speech Translation System Using Matcher Neural Networks, Proceedings of the Artificial Neural Networks in Engineering (ANNIE '93) conference, St. Louis, Missouri, 1993.

[27] Rumelhart, D.E., Hinton, G.E., and Williams, R.J., Learning internal representations by back-propagating errors, Nature, vol. 323, 533-536, 1986.

[28] Shirley, M.P., Coarticulation in American fingerspelling and its implications for automatic sign recognition, Master Thesis, University of Delaware, December, 1992.

[29] Shriberg, L.P., Kent, R.D., Clinical Phonetics, MacMillan Publishing Company, 1982.

[30] Stungis, J., identification and discrimination of handshape in American Sign Language, Perception 1 Psychophysics, 29 (3), 261-276, 1981.

[31] Tomkins, W. Indian Sign Language, Dover Publications inc, 1969.

[32] Virtual Technologies inc., CyberGloveTM User's manual, 1992.

[33] Wilbur, R.B., American Sign Language and Sign Systems, University Park Press, BA, USA, 1979.

[34] Woodward, J., Some observations on sociolinguistic variation in ASL, Kansas journal of sociology, 9(2), 191-199, 1973.