

AMERICAN SIGN LANGUAGE GENERATOR

APPROVED BY SUPERVISORY COMMITTEE:

Dr Prabhakaran Balakrishnan, Chair

Dr Kang Zhang

Dr Vincent Ng

Copyright 2008

Amruthraj Belaldavar

All Rights Reserved

AMERICAN SIGN LANGUAGE GENERATOR

by

AMRUHRAJ BELALDAVAR, B.E.

THESIS

Presented to the Faculty of

The University of Texas at Dallas

in Partial Fulfillment

of the Requirements

for the Degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

THE UNIVERSITY OF TEXAS AT DALLAS

August, 2008

ACKNOWLEDGEMENTS

Dr. Prabhakaran Balakrishnan has helped me immensely throughout my work and I want to thank him deeply for the same. I also thank him for the support he provided me during my courses at University of Texas at Dallas. Also I want to thank all the students at Animation and Graphics lab, especially Gaurav N. Pradhan, for their support and co-operation. And I also want to thank Dr. Yang Liu for guidance she provided during tweaking of Sphinx speech recognizer.

June, 2008

AMERICAN SIGN LANGUAGE GENERATOR

Publication No. _____

Amruthraj Belaldavar, M.S.
The University of Texas at Dallas, 2008

Supervising Professor: Dr. Prabhakaran Balakrishnan

American Sign Language (ASL) is a dominant Sign Language of the deaf community in the United States, in the English-speaking parts of Canada, in parts of Mexico and also in many other countries of the world. The idea behind this thesis was coming up with a tool that could Generate American Sign Language from the previous generated motions, as and when a person speaks the words. This would be a tool great of value for the teachers to teach the hearing challenged students. Through this thesis I have tried to come up with such a tool.

TABLE OF CONTENTS

Acknowledgements.....	vi
Abstract.....	vi
List of Figures.....	vi
CHAPTER 1 INTRODUCTION	1
CHAPTER 2 American Sign Language	2
2.1 History.....	2
2.2 Linguistics.....	2
2.3 Baby Sign.....	4
CHAPTER 3 American Sign Language Generator	6
3.1 CyberGlove.....	7
3.2 Sphinx-4.....	8
3.3 HMM-based Speech Recognition System	8
3.4 Java 3D.....	10
3.5 Humanoid Hand Model.....	11
3.6 Java Servlet Technology	12
3.7 Apache Tomcat	12
3.8 Jakarta HttpClient	12
CHAPTER 4 ASL Generator Implementation	13
4.1 Data Collection	13
4.2 Architecture of ASL Generator.....	13
4.3 Customizing Sphinx for ASL Generator.....	14
4.4 Implementation of Motions Driver	15
4.5 Web enabled ASL Generator	15
CHAPTER 5 Evaluation and quantitative results.....	17
5.1 System Requirements.....	17

5.2	Testing and Results	17
5.3	Bottlenecks.....	19
CHAPTER 6 Summary.....		20
6.1	Other possible implementations of an ASL Generator	20
6.2	Conclusions.....	20
CHAPTER 7 Suggestions for Future Work.....		21
REFERENCES		22
VITA		

LIST OF FIGURES

Number	Page
Figure 1. CyberGlove	8
Figure 2. An overview of main components of Sphinx-4	10
Figure 3. Humanoid Model	12
Figure 4. Architecture of ASL Generator	16
Figure 5. Accuracy Test Results	21

CHAPTER 1

INTRODUCTION

American Sign Language (ASL) is a dominant Sign Language for the deaf community in the United States, in the English-speaking parts of Canada, in parts of Mexico and also in many other countries of the world. It's a primary language for an estimated range of 500,000 to 2 million people in the U.S. alone. It is also the primary language of the hearing challenged students through which they learn other subjects. Currently, there is no tool in the teaching community that can be used as an aid to teach the hearing impaired students. Moreover, to teach hearing impaired students, an instructor requires proficiency in ASL. If there could be a tool that could generate a visual representation of ASL as and when the instructor speaks into, it will liberate the instructor from the need of knowing the ASL. Also it can become an invaluable tool for the hearing challenged students as they can use such a tool for independent learning by themselves whenever they can an audio version of the lecture or conference.

CHAPTER 2

AMERICAN SIGN LANGUAGE

2.1 History

Deaf educator Thomas Hopkins Gallaudet is credited with popularizing the signing technique in North America. At the behest of a father who was interested in education for his deaf daughter, Alice Cogswell, Gallaudet was enlisted to investigate methods of teaching the deaf. After visiting Paris and doing research he returned with Laurent Clerc, of Abbé de l'Épée's school, and together they founded the American Asylum for the Deaf and Dumb (now the American School for the Deaf), in Hartford, Connecticut, to teach sign language to American deaf students. It was at this school that all these influences would intermingle, interact and what would become ASL was born. The language continues to grow and change like any living language. In particular, ASL constantly adds new signs in an attempt to keep up with constantly changing technology.

2.2 Linguistics

ASL is a natural language and contains phonology, morphology, semantics, syntax and pragmatics just like spoken languages. It is a manual language or visual language, meaning that the information is expressed not with combinations of sounds but with combinations of hand shapes, palm orientations, movements of the hands, arms and body, location in relation to the body, and facial expressions. While spoken languages are produced by the vocal cords only, and can thus be easily written in linear patterns, ASL uses the hands, head and body, with constantly

changing movements and orientations. Like other natural sign languages, it is "three dimensional" in this sense.

2.2.1 Finger spelling

ASL includes both finger spelling borrowings from English, as well as the incorporation of alphabetic letters from English words into ASL signs to distinguish related meanings of what would otherwise be covered by a single sign in ASL. For example, two hands trace a circle to mean 'a group of people'. Several kinds of groups can be specified by hand shape: When made with C hands, the sign means 'class'; when made with F hands, it means 'family'. Such signs are often referred to as initialized signs because they substitute the first initial an English word as the hand shape in order to provide a more specific meaning.

2.2.2 Grammar

Phonology

ASL signs are divided into several phonemic features: hand shape, palm orientation, hand movement, hand location, as well as non-manual features such as facial expression, posture, and mouthing. Also signs are divided into segments of movement and hold, each of which consists of a set of the other features of hand shape, orientation, location, plus any non-manual features.

Morphology

ASL morphology is to a large extent iconic. Many spoken languages have both inflectional and derivational morphology. ASL appears to have only derivational morphology. There are no

inflections for tense, number, or person. Person is indicated indexically with some verbs, but the form this takes is specific to each verb, and can't be arbitrarily extended to new verbs the way verbal inflections can. A similar situation exists with verbal number.

Syntax

ASL syntax is primarily conveyed through a combination of word order and non-manual features

2.2.3 Writing systems

ASL is often written with English words in all capital letters, which is known as glossing. This is, however, a method used simply to teach the structure of the language. ASL is a visual language, not a written language. There is no one-to-one correspondence between words in ASL and English, and much of the inflectional modulation of ASL signs is lost.

There are two true writing systems in use for ASL: a phonemic Stokoe notation, which has a separate symbol or diacritic mark for every phonemic hand shape, motion, and position (though it leaves something to be desired in the representation of facial expression), and a more popular iconic system called SignWriting, which represents each sign with a rather abstract illustration of its salient features. SignWriting is commonly used for student newsletters and similar purposes.

2.3 Baby Sign

In recent years, it has been shown that exposure to sign language has a positive impact on the socialization of hearing children. When infants are taught to sign, parents are able to converse

with them at a developmental stage when they are not yet capable of producing oral speech, which requires fine control of both breathing and the vocal tract. The ability of a child to actively communicate earlier than would otherwise be possible appears to accelerate language development and to decrease the frustrations of communication.

CHAPTER 3

AMERICAN SIGN LANGUAGE GENERATOR

This is the tool implemented as part of this thesis which generates visual representation of ASL as and when a person speaks the words. It has been implemented using Open Source Software's. Different tools used in building ASL Generator are listed as below:

- CyberGlove – Motion Capture data glove developed by Immersion Corporation
- Sphinx – A speech recognizer written entirely in the Java programming language
- Java 3D – Animation API from Sun Microsystems
- Humanoid Hand Model - Model developed to simulate the kinematics of a human hand
- Java Servlet Technology from Sun Microsystems
- Apache Tomcat - An implementation of the Java Servlet technology by Apache Software Foundation.
- Jakarta HttpClient - Low-level libraries of the HTTP protocol developed by Apache Software Foundation.

3.1 CyberGlove

CyberGlove was used as Motion Capture Hardware during the implementation. CyberGlove is a fully instrumented glove that provides up to 22 high-accuracy joint-angle measurements. It uses proprietary resistive bend-sensing technology to accurately transform hand and finger motions into real-time digital joint-angle data. Each sensor is extremely thin and flexible being virtually undetectable in the lightweight elastic glove.

Sensors Distribution:

- Flexion Sensors per finger: 3
- Abduction Sensors: 4
- Palm-Arch Sensor: 1
- Wrist Flexion Sensor: 1
- Abduction Sensor: 1

The basic CyberGlove system comes with one CyberGlove, its instrumentation unit, a serial cable to connect it to host computer, and an executable version of VirtualHand graphic hand model display and calibration software. It also comes with VirtualHand Studio software that converts the data into a graphical hand which mirrors the subtle movements of the physical hand.



Figure 1. CyberGlove (Courtesy: vrlogic.com)

3.2 Sphinx-4

Sphinx-4 is a state-of-the-art speech recognition system written entirely in the Java programming language. It was created via a joint collaboration between the Sphinx group at Carnegie Mellon University, Sun Microsystems Laboratories, Mitsubishi Electric Research Labs (MERL), and Hewlett Packard (HP), with contributions from the University of California at Santa Cruz (UCSC) and the Massachusetts Institute of Technology (MIT).

3.3 HMM-based Speech Recognition System

Sphinx-4 is an HMM-based speech recognizer. HMM stands for Hidden Markov Models, which is a type of statistical model. In HMM-based speech recognizers, each unit of sound (usually

called a phoneme) is represented by a statistical model that represents the distribution of all the evidence (data) for that phoneme. This is called the acoustic model for that phoneme. When creating an acoustic model, the speech signals are first transformed into a sequence of vectors that represent certain characteristics of the signal, and the parameters of the acoustic model are then estimated using these vectors (usually called features). This process is called training the acoustic models.

During speech recognition, features are derived from the incoming speech in the same way as in the training process. The component of the recognizer that generates these features is called the front end. These live features are scored against the acoustic model. The score obtained indicates how likely that a particular set of features (extracted from live audio) belongs to the phoneme of the corresponding acoustic model.

3.3.1 Sphinx-4 Architecture and Main Components

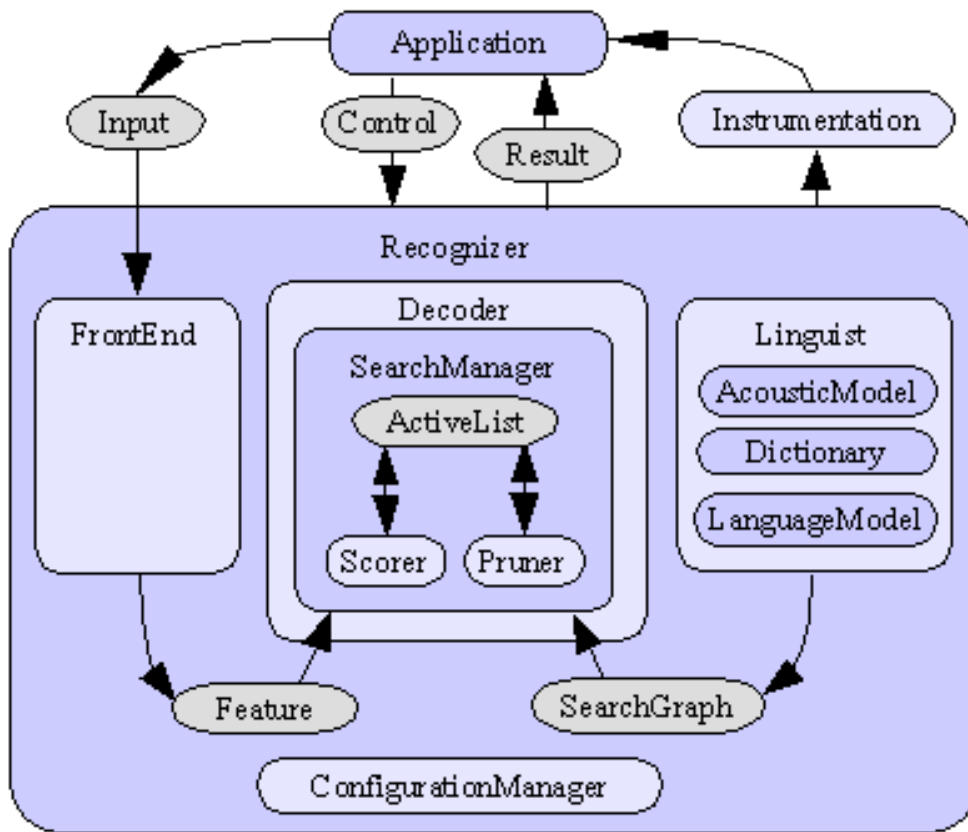


Figure 2. An overview of main components of Sphinx-4 (Courtesy: cmusphinx.sourceforge.net)

3.4 Java 3D

The Java 3D API is a hierarchy of Java classes which serve as the interface to a sophisticated three-dimensional graphics and sound rendering system. Java 3D provides high-level constructs to create and manipulate 3D geometry, and to build the structures used to render that geometry.

Using its API, developers can efficiently create precise virtual universes in a wide variety of sizes, from astronomical to subatomic.

3.5 Humanoid Hand Model

It's a model developed to simulate the kinematics of a human hand. It's a Pure Java based Open Source project developed by Stefan Hendrickx as part of his Robotics coursework at K.U.Leuven

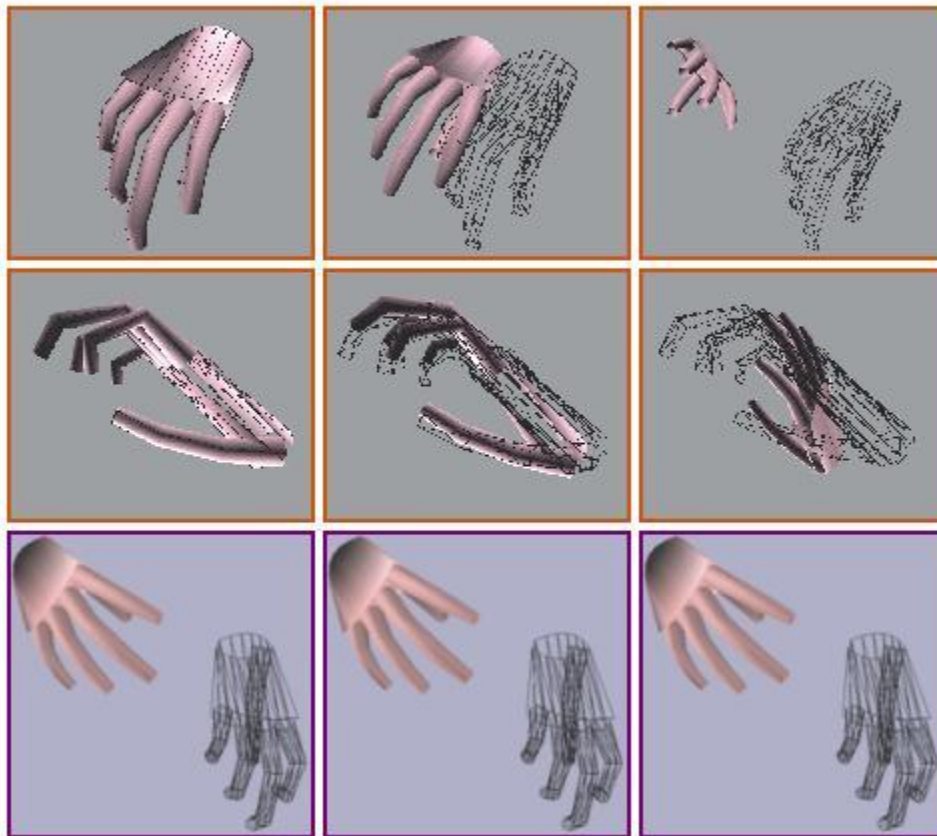


Figure 3. Humanoid Model (Courtesy: Stefan Hendrickx,

<http://www.geocities.com/stefanhendrickx/hand/robotics.html>)

3.6 Java Servlet Technology

Servlets are the Java platform technology for extending and enhancing Web servers. Servlets provide a component-based, platform-independent method for building Web-based applications, without the performance limitations of CGI programs. Servlets are server and platform-independent.

Servlets have access to the entire family of Java APIs. Servlets can also access a library of HTTP-specific calls and receive all the benefits of the mature Java language, including portability, performance, reusability, and crash protection.

3.7 Apache Tomcat

Apache Tomcat is an implementation of the Java Servlet and JavaServer Pages technologies. Apache Tomcat has been developed in an open and participatory environment and released under the Apache Software License. Apache Tomcat powers numerous large-scale, mission-critical web applications.

3.8 Jakarta HttpClient

Jakarta HttpClient is a low-level library of the HTTP protocol developed by Apache Software Foundation. HttpClient enable building of HTTP-aware client and server applications such as web browsers, web spiders, HTTP proxies, web service transport libraries, or systems that leverage or extend the HTTP protocol for distributed communication. Users can build custom HTTP services with advanced needs in server- or client-side HTTP communication.

CHAPTER 4

ASL GENERATOR IMPLEMENTATION

4.1 Data Collection

ASL Generator requires hand motions to be generated, cleaned and maintained before it can be used in the tool. For generating the motions, CyberGlove was used.

Steps involved in generating motions using CyberGlove:

- Calibrating CyberGlove
- Generating the motion
- Collecting the data
- Cleaning up of data to suit the needs of ASL Generator

The cleaned up files are stored as comma separated values (.csv) files

4.2 Architecture of ASL Generator

Architecture of the ASL Generator consists of 5 different components. They are as follows:

- Sphinx Speech Recognizer
- Model Driver
- HTTP Client
- HTTP Server
- Motions Database

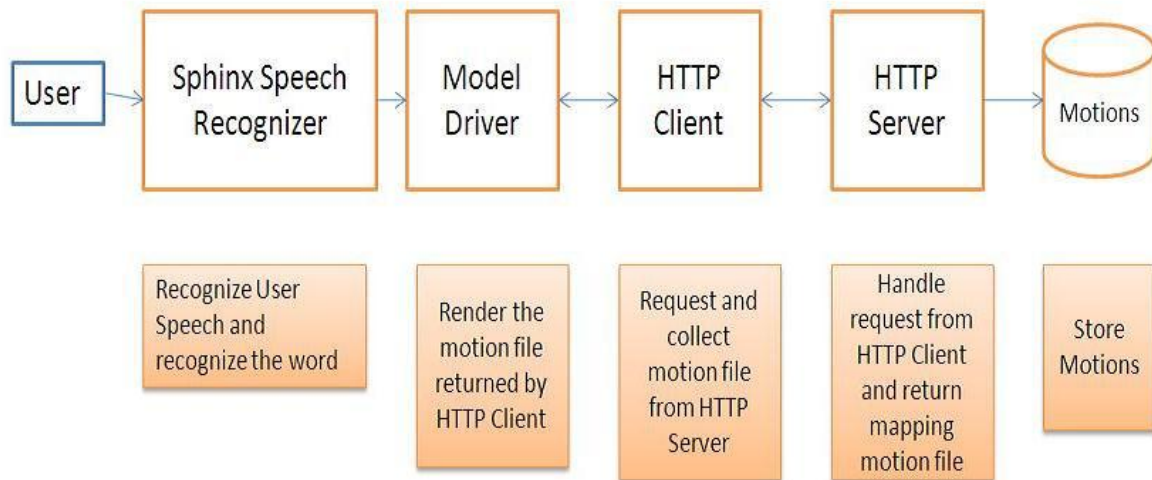


Figure 4. Architecture of ASL Generator

Speech Recognizer listens to the user for speech, converts the spoken words into text and then passes on the recognized word to the Model Driver as a parameter. Model driver created a HTTP Client and passes the word for which motion information needs to be fetched from the Server. The HTTP Client talks to the HTTP Server component running on the central server and which in turn fetches the corresponding motions file from Motions Database and returns to HTTP Client. HTTP Client returns the same to Motions Driver. Now with the data file retrieved from the server, Motions Driver renders the motion to the user. This interaction between different components of ASL Generator is shown in the figure above.

4.3 Customizing Sphinx for ASL Generator

Sphinx comes up configuration files to customize it for the needs of the application in which it is being used. These configurations include Word Recognizer, Decoder, Linguist Model, Grammar, Dictionary and Acoustic Model. All these configurations go into a file called

signlanguage.config. And the grammar for the words to be recognized by sphinx is specified in signlanguage.gram file. A component that uses Sphinx for speech recognition is implemented in SignLanguage.java.

4.4 Implementation of Motions Driver

Motions Driver is at the heart of the ASL Generator. It acts as the controller of the system as well as the component that renders the motions on the screen for the user.

Its functions include:

- Act as the controller of the tool. It passes on the data between speech recognizer component and the HTTP Client.
- Render motions data obtained from HTTPClient
- Provide Synchronization during motions rendering

Synchronization performs the following operations:

- Starting a new motion after the end of a motion
- Smother transition between 2 consecutive motions
- Buffering of words spoken faster than the speed of display of the motion in the model

4.5 Web enabled ASL Generator

For the tool to be effective it needs to contain a good collection of words of ASL. That means a large collection of .csv files. This poses the challenge for easy distribution of the tool. One approach of dealing with this situation is storing all the motions files in a central database and

letting the users' access these files using the ASL Generator client easily downloadable onto their systems.

So with this architecture:

- All of the data files (.csv motion files) remain on the server
- Only the Model Driver can be downloaded by the user
- As and when the words are spoken, the data files are fetched from the central server and displayed by the Model Driver

This approach also provides other advantages:

- Updated motions files can be uploaded very easy at the central server repository without requiring the users to periodically updating their motions databases.
- More words can be easily uploaded into the central server

CHAPTER 5

EVALUATION AND QUANTITATIVE RESULTS

5.1 System Requirements

Following are the system requirements for ASL Generator:

5.1.1 Software Requirements:

ASL Generator can run on either Windows or UNIX and Linux based systems

- Sun Microsystems JRE 1.4 or above
- Installation Java 3D API
- An implementation of the Java Speech API (A requirement of Sphinx 4)
- Apache Tomcat 5.0 or above

5.1.2 Hardware Requirements:

- A good working Microphone
- A Processor with speed of 1 GHz or more
- 1GB or more of RAM

5.2 Testing and Results

ASL Generator has been tested on multiple systems meeting the above system requirements and the results were found to be as expected. The tool behaved consistently.

5.2.1 Running the tool:

Below steps have been mentioned with respect to the ASL Generator available through the archive (.zip) format

5.2.2 Setup:

As mentioned in the system requirements section above, the tool requires Java3D to be available on the system for execution. If Java3D is not already installed in the system, following the step below installs Java 3D on the system. Otherwise one can skip the step and directly run the tool.

Copy the 3 dll files (j3dcore-d3d.dll,j3dcore-ogl-cg.dll,j3dcore-ogl.dll) in dll-files folder into the bin folder of JRE. eg.. If the JRE is installed at C:\Program Files\Java\jre1.5.0_08\ then copy the 3 dlls into C:\Program Files\Java\jre1.5.0_08\bin directory.

5.2.3 Running:

1. Connect the microphone to the system
2. In the extracted folder, browse to bin directory (ie, signlanguage\bin) and click on run.bat
3. When the program starts and as you speak the first word, the window of Hand opens up. Click on the hand and press any key to activate the motion of Hand
4. Enter Control-C to terminate the program

5.2.4 Results:

Below are the results of tests run on system with 1.6GHz processor and 2GB of RAM.

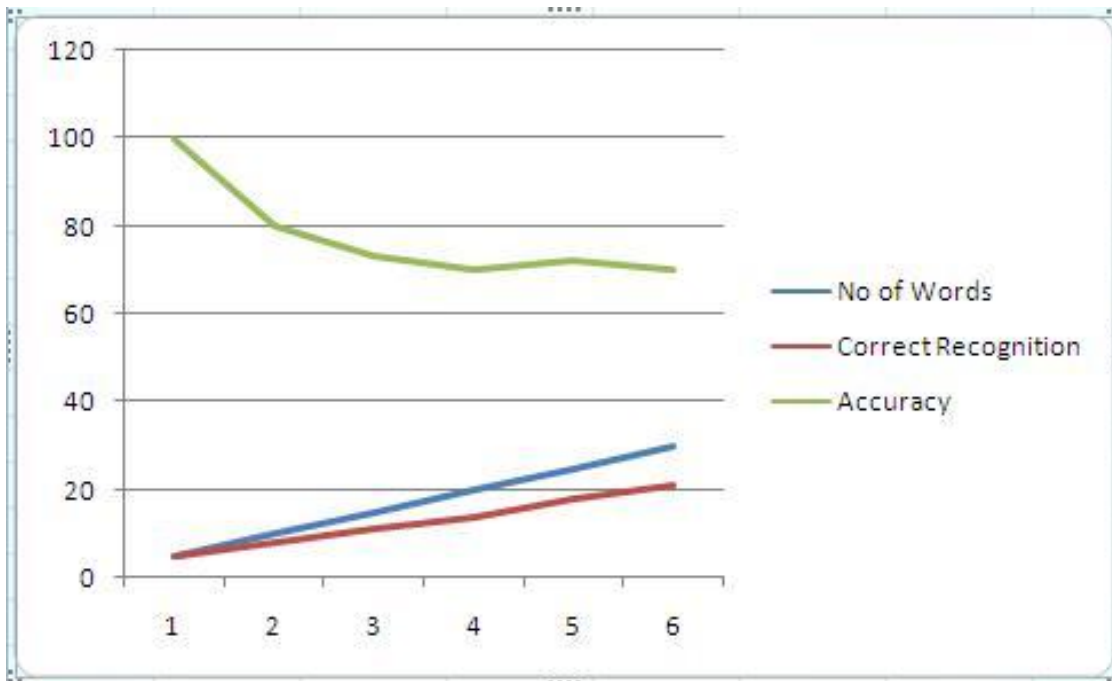


Figure 5. Accuracy Test Results

5.3 Bottlenecks

As visible from the test results in the section above, the speed and accuracy of the sphinx becomes a bottleneck of the tool.

CHAPTER 6

SUMMARY

6.1 Other possible implementations of an ASL Generator

One of the possible approaches for implementing a similar ASL Generator would use pre-generated videos of the ASL words and then use them for the display. However generating video for continuous speech would be impossible due to huge number of possible sentences that can be spoken in real word. Also it becomes very difficult to provide smoother transition between different words when videos are used for display of ASL motions.

6.2 Conclusions

ASL Generator could be used as a tool by the ASL unaware teachers to teach subjects to hearing challenged students. Also many special purpose applications can be developed with ASL Generator as the model. E.g.,

- Telecasts and live Reports
- Announcements at public places

CHAPTER 7

SUGGESTIONS FOR FUTURE WORK

With this implementation of ASL Generator as a model many other applications can be developed. One of possible future work could consider handling continuous speech. Providing support for continuous speech would make this tool for meaningful and comprehensive. It would open the way for the tool being used in real applications.

Also integrating the application with a faster and more accurate Speech recognizer than Sphinx would improve the usability of the tool significantly.

REFERENCES

- Gaurav N. Pradhan, Chuanjun Li, Balakrishnan Prabhakaran, “Hand Gesture-based Computing for Hearing and Speech Impaired”, IEEE Multimedia Magazine, Vol. 15, No. 2, pp. 20-27, April-June 2008.
- American Sign Language browser
<http://commtechlab.msu.edu/sites/aslweb/browser.htm>
- Wikipedia - American Sign Language <http://www.wikipedia.org>
- Documentation and examples provided at Sphinx Website
<http://cmusphinx.sourceforge.net/sphinx4/>
- Immersion Corporation Website www.immersion.com/3d/products/cyber_glove.php
- Stephan Hendix's Website
<http://www.geocities.com/stefanhendrickx/hand/robotics.html>
- Sun Microsystems Website <http://java.sun.com/>
- The Apache Software Foundation <http://www.apache.org>

VITA

Amruthraj Belaldavar obtained his B.E. from Sri Jayachamarajendra college of Engineering, Mysore, India in 2002. Since then, he has developed extensive Professional experience in the field of Internet Applications specializing in Enterprise Applications. He has worked for different Multinational Organizations developing Software Services and Products for them. He is currently working at Sabre Airline Solutions at Southlake, TX working on Python based Web Robots. He has been a member of IEEE and Computer Society of India in the past and is currently an active member of Association of Computing Machinery.