Sign Language Translator using Microsoft Kinect

A Dissertation submitted towards the partial fulfillment of
the requirement for the award of degree of

Master of Technology
in
Signal Processing & Digital Design

Submitted by

Mohd Shamshad Ansari

2K15/SPD/08

Under the supervision of

Dr. Dinesh K. Vishwakarma
(Assistant Professor, Department of ECE)

Department of Electronics and
Communication Engineering

Delhi Technological University

(Formerly Delhi College of Engineering)
Shahbad, Daulatpur - 110042

July, 2017

DELHI TECHNOLOGICAL UNIVERSITY
Established by Govt. Of Delhi vide Act 6 of 2009
(Formerly Delhi College of Engineering)
SHAHBAD DAULATPUR-110042

CERTIFICATE

This is to certify that the dissertation title "***Sign Language Translator using Microsoft Kinect***" submitted by *Mr. Mohd Shamshad Ansari*, Roll. No. *2k15/spd/08,* in partial fulfillment for the award of degree of Master of Technology in "Signal Processing and Digital Design (SPDD)", run by Department of Electronics & Communication Engineering in Delhi Technological University during the year 2015-2017, is a bonafide record of student's own work carried out by him under my supervision and guidance in the academic session 2016-17. To the best of my belief and knowledge the matter embodied in dissertation has not been submitted for the award of any other degree or certificate in this or any other university or institute.

Date:

Project Guide:

Dr. Dinesh K. Vishwakarma

Assistant Professor

Deptt. of Electronics & Comm. Engineering

Delhi Technological University,

(Formerly Delhi College of Engineering)

Delhi -110042

-

DELHI TECHNOLOGICAL UNIVERSITY

Established by Govt. Of Delhi vide Act 6 of 2009

(Formerly Delhi College of Engineering)

SHAHBAD DAULATPUR-110042

DECLARATION

I hereby declare that the work presented entitled "**Sign Language Translator using Microsoft Kinect**" as Dissertation in requirement of partial fulfillment for the award of the degree of Master of Technology in Signal Processing and Digital Design and submitted to Delhi Technological University, Delhi is an authenticate record of my own work carried out under the supervision of **Dr. Dinesh K. Vishwakarma**, Assistant Professor in the department of Electronics & Communication Engineering .

The matter presented in this thesis has not been submitted by me for the award of any other degree of this university or any other university or institute.

Date:

Mohd Shamshad Ansari

M. Tech. (SPDD)

2k15/spd/08

ACKNOWLEDGEMENT

First of all, I would like to express my gratitude to my advisor, **Dr. Dinesh K. Vishwakarma,** Assistant Professor, Department of Electronics & Communication Engineering**.** He has been present throughout my entire journey at Delhi Technological University, as the professor in my very first class here, to the final signature on my thesis. I truly admire his depth of knowledge and strong dedication to students and research, that has made him one of the most successful professors ever. He has provided me tremendous insight and guidance throughout my research work, and I'm not sure where I would be without his encouragement and moral support. I am glad that I was given opportunity to work with him.

I am greatly thankful to entire faculty and staff of electronics & Communication Engineering Deptt. for their, continuous support, encouragement and inspiration in the execution of this "**Dissertation"** work.

Finally, I want to thank my parents, family and friends for always believing in my abilities and showering their invaluable love and support.

Mohd Shamshad Ansari

2k15/spd/08

ABSTRACT

In any community there are people who face severe difficulties in communication due to their speech and hearing incapability. Such people use a number of gestures and symbols to convey and receive their messages and this form of communication is called Sign Language. On the other hand a natural language speakers do not understand the sign language , resulting in a communication hindrance amongst the people weakening their social interaction. To minimize this communication gap, there is a need to develop a system which can consists of two independent modules i.e one which translates sign/gesture into text/speech and second which translate speech into sign/gesture. For this purpose we have provided solution based on dynamic time warping for the first module and a software based solution for the second module by exploiting latest technology of Microsoft Kinect depth camera which trackes the 20 joint location of human beings. In sign to speech/text conversion block, the actor perform some valid gestures within the kinect's field of view. The gestures are taken up by the kinect sensor and and then interpreted by comparing it with already stored trained gestures in the database. Once the gesture is recognized it is mapped to the respective word which is sent to the text/speech conversion module to produce the output. In the second block, which is speech to sign/gesture conversion, the person speaks in kinect's field of view which is taken by the kinect and the system converts speech into text and corresponding word is mapped into predefined gesture which is played on the screen. This way a disabled person can visualize the spoken word. The accuracy of sign to speech module is found to be 87% and that of speech to gesture module is 91.203%.

Table of Contents

List of Figure

List of Table

Chapter 1

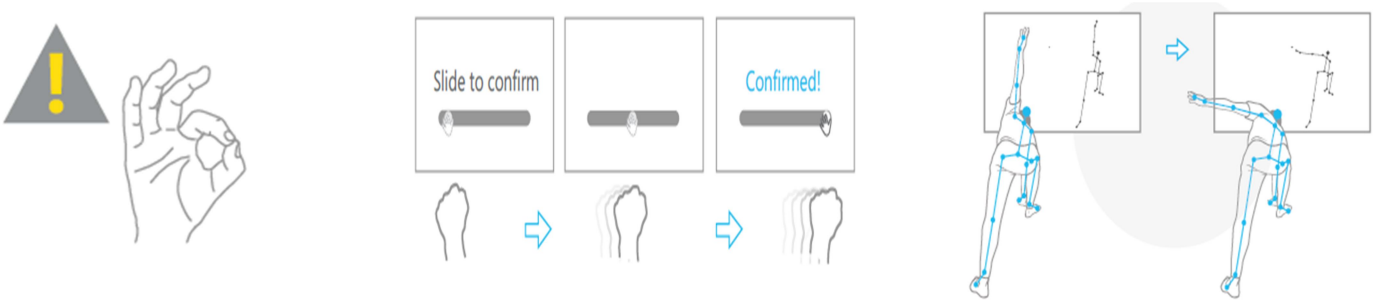
1 Introduction

In our world, there exist people who don't have the hearing and speaking abilities, and such people use a sign language as the main means of communication. Sign language involves the use of hands gestures, the head, body posture and facial expression. To facilitate the exchange of information between normal and hearing impaired people, either the normal person has to learn the sign language and its meaning or the third person who knows both the language can be used to act as a translator between the two person. The other alternative is to use the computer between the two person , signer and non-signer as a translator and interpreter. Ideally the computer will take the sign/gesture performed by the user in front of it and interpret it and convert it into speech/text output for the normal person, the system should also listens to the normal person and maps it to the gesture/text for the hearing impaired person to understand. Such a system has been developed here for limited words in the dictionary.

Though the sign language has been used for centuries but its standardization began in seventeenth century onwards. As natural language has its own morphological organization and grammatical nuances so as the sign language. It is found in variety of forms. In case of Finger-spelling or manual signing , a sequence of 'alphabet signs' makes a word and to make a sentence recursion is used. The advantage of manual signing is that it does not require a wide vocabulary. But for the traditional and mostly used systems have almost every word in the sign dictionary. Although there are a lot of variations in these signs from region to region but a few standardised systems have came up in the past such as French sign language, British sign language, American sign language etc.

The term gesture can be defined as any form of movement that can be used as an input or any interaction to control an application. It can take various forms such as simple hand pose, moving a specific pattern using hand and long stretches of continuous movement using the whole body.

Signs can be categorized into three parts static signs, dynamic signs and continuous sings. The static signs involve fixed hand pose and can be defined as the pose or posture which the user must match and application recognize it as meaningful. For example: symbol used for "Okay" see the figure 1.1(a) . In case of dynamic gestures, the continuous movement is there which allows user to directly manipulate the object or control and receive continuous feedback. For example: "Pressing to select" and "gripping to move" figure 1.1(b). Lastly in the continuous gesture, prolonged tracking of the movement is there having no specific pose but the movement is used to interact with the application. Example include enabling the user to start the virtual box where whole body movement is performed [2], figure 1.1(c).



(a) Static sign (b) Dynamic sign (c) Continuous sign

Figure 1.1: Different gestures to control Application

In a survey done by World Health Organization (WHO), it was found that around 125,000 human infants are born deaf-mute every year all around the world and 0.36 billon people got hearing loss problem. It is estimated that it covers around 5.4% of the globe population and out of that 90% are adults. To help people suffering form such disabilities, a lot of research has been conducted in the past and some solutions[15] have been made but no big success is reported so far [1] (Mateen Ahmed, 2016).

The goal of this thesis is to develop an automatic Sign language Translator that can provide the people who are suffering from speaking and hearing problem, a natural way to communication with the people all around. The system developed will be two way translator providing dual mode of communication such that when mute person does some sign means providing the system sign language it translates that into text and speech as natural language

whereas when a person speaks the system takes the input as a natural language , finds its corresponding gesture and show the gesture in sign language . The presented system is based on the latest technology of the “Kinect” which follow the motion of human, provides gesture and depth image. This way developed system can provide the people who suffers hearing loss and speaking disabilities, a helping hand to overcome their loss of hearing and speaking.

Our experiments show that the discriminative nature proposes us to develop the prototype to detect the movement of the body parts and joints with a Kinect sensor and these functions are capable enough to be used as a standard deterministic optimization method, which needs very less computation and can however provide excellent results even in adverse environment.

This kind of system can provide new ways for employement to the speaking and hearing challenged people especially at tourist confirmation counter and hospitals. Imagine an information kiosk, say, at an hospital, and rather than the person seeking information being deaf, imagine that the person staffing the information kiosk has speaking and listening incapability as shown in Figure 1. Now, a non-signer could come to that kiosk and ask questions and could use the system to help them communicate. This system can also be implemented in the universities itself, for instance, in the lecture halls in Delhi Technological University (DTU) to break the boundaries between the lecturers that were deaf or mute and the students. It can also be applied to teleconferencing via Sign Language Interpreter System to enhance the system for users to fully utilize the facilities to the maximum level.

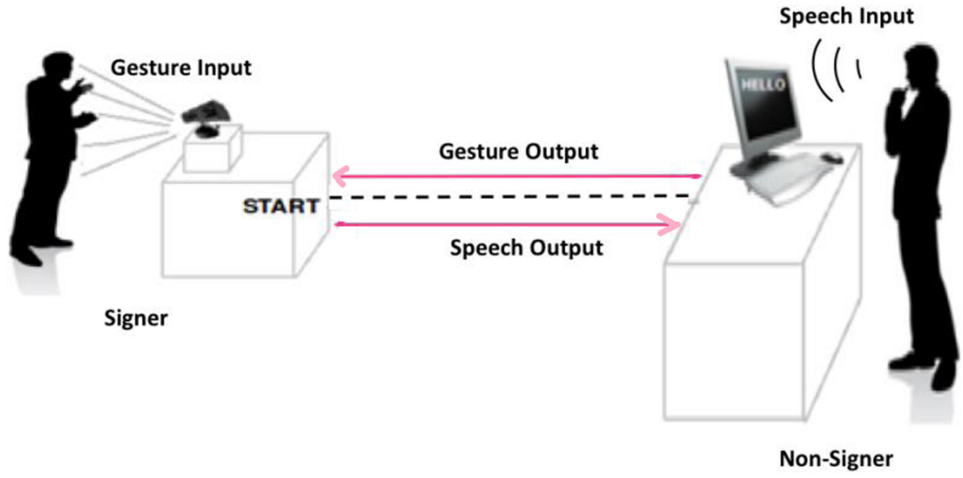


Figure 1.2: Two-ways communications between the signer and the non-signer

The two modules of sign language translation i.e. sign to speech translation and speech to sign translation. The tools and techniques along with the components used have been described in detail, giving a complete technical and logical insight of the methodology for each of the modules. Towards the end, experiments are performed on the system for both the modules and accuracy of the system is explained which helps evaluating the goodness of the proposed approach and obtained the results.

1.1 Thesis Proposal

The proposed system implements the two modules of the Sign Language Translation i.e. sign to speech translation and speech to sign translation.

The first module of the proposed Sign Language Translator which is sign to speech/text converter satisfies the following goals:

- Provide an natural way of the communication between speech impaired and normal person
- Use the data provided by the Microsoft Kinect Xbox 360® such as the joint location coordinates
- Recognize basic signs stored in the dictionary. This dictionary contains 10 basic signs which can be performed by the signer in front of the kinect camera.

• **Right Hand Swipe right**	• **Right Hand Swipe Left**
• **Left Hand Swipe Right**	• **Left Hand Swipe Left**
• **Right Hand Wave**	• **Left Hand Wave**
• **Both hands push up**	• **Two hands zoom out**
• **Right hand push up**	• **Left hand push up**

Table 1.1: Dictionary of signs used of the system

- Provide an interactive interface so that the user can run the application easily and without any in depth knowledge about the software.

- The system enables real time user interaction providing instantaneous output as soon as the sign is performed.

The second module of the proposed sign language translator which is speech to sign converter satisfies the following goals:

- Exploit the hearing capability of the Microphone array of the kinect to detect the person's voice.
- the basic words whose human gestures will be produced as output are

Hello	Good Morning
How are you	Happy
One	Two
You	I/Me
Left	Right
Three	Four
Five	Fine

Table 1.2: spoken words collection for module 2.

Besides these main goals of the project it also deals with some hidden task.

- Becoming familiar about Microsoft kinect Xbox 360® and how to use depth data it provides.
- Use of .NET framework and developing platforms such as Microsoft Visual Studio based on C#.
- Download and install Microsoft SDK and other tools to work with Microsoft Kinect Xbox 360.
- Becoming familiar with Dynamic Time Warping Algorithm and classifier such as Nearest Neighbor classifier.
- Study about different feature descriptors to describe the signs.

- Learn about Microsoft visual Studio window to design Sign Language Translator user interface.
- Describe the human gesture for different signs

The main aim of the thesis is to exploit the latest technology of the Microsoft kinect Xbox 360® with the combination of a basic descriptor and classifier for the first module of Sign Language Translation task which is sign to text conversion. A software based solution has been developed for second module of the sign language translator which is speech to sign/gesture conversion. The system must be able to detect a wide number of signs done by users of different sizes.

1.2 Overview of the thesis

The whole thesis is divided into four different chapters. The *chapter 2*: *Background,* discusses about related work which is necessary to understand the project. Next, Basic knowledge of the Sign languages, different sign capturing methods (which discusses about Microsoft Kinect), system requirement is discussed. In *chapter 3: Methodology,* the methodology and steps which are followed to implement the system is given. The *chapter 4: Experiments and Results,* gives the user real time interaction with the system and experiments are done with different testers to evaluate the implemented system and results are shown. Finally, In *chapter 5: Conclusion*, the thesis is summarized and initial goals are checked. To wrap up the thesis, *Appendix A* provides the dataset of the signs stored in the dictionary.

Chapter 2

2 Background

In this section of background, first of all, the related work is overview. Then, the basic knowledge about Sign Language and the Microsoft Kinect Xbox 360 is given and finally the software technology and architecture is studied.

2.1 Related work

In the field of sign language, previous works mainly focused on sign language recognition systems. They can be arranged into the action recognition field, which is a complex task that involves many aspects such as motion analysis, pattern recognition, motion modeling, machine learning, and even sometimes psycholinguistic studies. Some of the works have been done before on Human Action Recognition [4],[5] and [6] but they were based on 2-D information and a few based on depth data(3-D).

Yang Quan et al. discussed a Basic Sign Language Recognition system. This system is able to translate a sequence of human signs into the commonly used speech language and vice versa [8]. This system which is a bidirectional (sign to speech and speech to sign) allows the deaf and mute people to communicate effectively, focused on the Chinese Manual Alphabet where every sign belongs to a letter from the alphabet. This system was thought to be installed in public places such as airports, railways and hospitals. The system is made up of a camera, a speaker, a video display terminal, a mic and a keyboard. It utilized two kinds of data: vectors of lip actions and hand gestures. To characterized these vectors, they used Hu moments [10] and the Normalised Moment of Inertia algorithm [9]. The former is a set of algebraic invarients. It combines regular moments that provides shape information of the image. The latter makes the system invariant of the translation and size. As said before, it combined the hand gesture recognition with lips movements to make the system more robust. It used multi feature SVM classifier which is trained by the linear kernel. The system recognized the 30 letters from the Chinese alphabets with an accuracy of 95.55%.

A view-based approach was presented by Starner et al. [13] that uses a single camera to extract 2-D features which is given as the input of Hidden Markov Model for continuous American Sign Language Recognition. They obtained word accuracy of 92% in recognizing the sentences with 40 different signs.

Some other projects such as [11] make use of data gloves where every finger contained a different color. In [11], a sign language translation system is introduced which extracts the feature using color segmentation and uses Neural network as a classifier which can detect letters(A-Z), numbers(1-9), and some words upto 12. In case of letters and numbers, it defined a 10-array vector that contains x and y offsets representing the distance between each finger and centroid of the hand. For words, it avoided finding position of the fingers and only concentrated on the centroid of the hand. Hence, those sequences which finds the centroid similar gives the frame of the same sign. Finally, neural network classifiers are used which were trained with the signs in the dictionary. The average accuracy of the system was obtained as 96.668%.

A HMM-based gesture recognition system was demonstrated by Jonathan C. Hall in [12], a good solution for the 3D data. Any physical sign can be taken as Markov chain where the true state is not known directly. This type of markov model is called Hidden Markov Model. To reduce the real gesture data to a limited number k-means was used to cluster the coordinates of every gesture.

A parallel hidden Markol Model base solution was introduced by Vogler et al. in [13] which used 3-D data as input. In American Sign Language, modeling process of the signs becomes complex because of the fact that the combination of the phonemes are large. They have created a framework which takes the 3D input and classify the sign. The 3D data is obtained from 3D computer vision methods or with a magnetic tracking system such as Ascenion Technologies Motion Star system. They showed how to apply this framework in practice with 22-sign vocabulary. The test accuary found to be 95.83%.

2.2 Sign Language

In sign language communication which is a manual communication the user can use facial expression, hand shapes and its motion and orientation, arms or body and its movement to express his/her thought. Sign language have many similarity with oral (spoken) language, which primarily depends on the sound. Both are considered as natural languages by the linguists because of the fact that they share same linguistic properties and language faculties. Both are different than body language which is a non linguistic language.

Sign language is not only used by the deaf people it is also used by the people who can hear well, but have trouble in speaking. Sign language had developed wherever communities of such people exited so it is not clear that how many sign languages are there. The Ethnologue listed 137 sign languages in its 2013 edition. A sign language may be native to one place and can't be recognized at other places. Not only that its standardization began seventeenth century onwards. Now we have some standard sign languages such as ASL,BSL etc.

Depending on the need and situation, the signer can use one hand or both hands to perform the sign in the signing space. It can have a particular orientation and location. If it changes the sign will change.

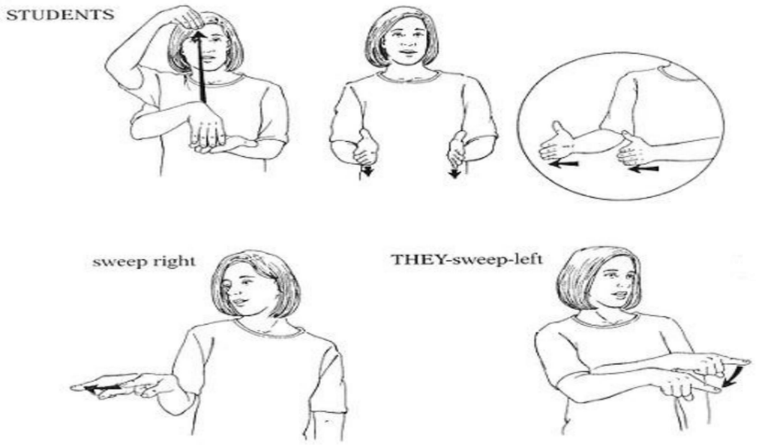


Figure 2.1: Hand shape distinctive features

The goal in the section is to give the idea of sign language's basic structure (morphology, phonetics and syntax). The grammar rules are defines by group of people who use the sign language in a particular region, so we find a lot of variation in these

rules. When a number of people use a sign rule often enough that using it in any other way seems odd, a new grammar rule come into existence. So here we will not point to a particular sign language and its description but a general description for a group of sign languages sharing common characteristics. The attempt here is to make people who are not familiar with complexity of the sign language to realize how sophisticated it would be to design such an Sign Language Translator and the reason to reduce complexity of the system by not taking into consideration these characteristics was made in this version of system presented here.

Similar to spoken language, the sign language has its own grammar rules of morphology, phonology, syntax.

2.2.1 Morphology

Morphology is the study of morphemes. A morpheme is a part of the language which is meaningful and cann't be further broken down into independent parts but if it is divided into smaller parts they don't have independent meaning.

There are three types of morphemes:

(1) Free Morphemes

(2) bound Morphemes

(3) derivational Morphemes

Free Morphemes

These types of morphemes are meaningful themselves and don't require to be attached to some other words to define their meaning. They lost their existance if they are further divided. For example: the sign "happy" is a free morpheme, it cann't be broken down into meaningful parts.

Bound Morphemes

These types of morphemes need to be attached to some other morphemes to have their meaning ,they have no meaning if they are left independent. For example: the sign for plural of dog i.e. dogs, addition of "s" changes the meaning to its plural but if sign for "s" is performed independently it has no meaning.

Derivational morphemes

Derivational morphemes, if attached to a word it changes the meaning of the part of speech of the word. For example: we have sign for word “sit” but if we want a sign for word “chair” we will add some movement which changes the meaning. This is derivational morpheme.

2.2.2 Phonology

It is the study of phonemes which is the smallest unit of the spoken language. Similarly the phonemes for the sign language can be defined which comprises of following four parameters.

1. Hand shape: The shape of the hand defines the phonemes. if the shape of hand changes the phoneme changes. For example: words “can” and “now” have same hand shape but different movement.
2. Movement: the movement of the hand also defines the important parameter for the phoneme. For same hand shape, if the movement changes the word changes. For example: sign for words “sit ” and “train” have same movement of the hand.
3. Location: the location of the hand has a very significant role in defining the phoneme. The words having same hand shape but different locations have different meaning. For example: words “remember” and “girl” both have same hand shape but different location.
4. Palm Orientation: the palm orientation is also such a parameter which defines a phoneme. For example : the words “want” and “freeze” have different palm orientation.
5. Plane: On static position if the hand is in one plane, the plane changes if the movement of the hand is large.

2.2.3 Syntax

The principles and the rules which are used to form a sentence is known as syntax. A syntax may be defined as the study of constructing the sentence. In sign language the syntax is conveyed through the word order and non manual markers.

Word order:

Word order shows the order in which one can sign the words. Generally words are arranged in TOPIC-COMMENT form which is same as subject-predicate form similar to English language. for example: " I am a student" can be signed as "I STUDENT"

Sentence types

Sentence types are not word types. Infact sentence types provide us a way to use word types along with non-manual markers to form new types of sentences.

Questions

There are two types of questions used.

1. "wh" word questions
2. Yes/no questions

"**wh**"**question type**

Questions start with when, why, who, what and where. These "wh" words are signed at the end of the question. The non-manual markers used are

- Lean head forward and lower the eyebrow
- Hold the last sign in your sentence

Yes/no question type

This type of question requires only yes or no answer. The non-manual marker are

- Lean head forward and raise the eyebrow
- Hold the last sign in your sentence

Negation

To form the negative one can use "NOT" before the word and shake the head while signing. You can frown while signing the word. We can also reverse the orientation of the sign to form negative. Non manual markers are very significant part of negation. For example the sign for "DON'T WANT" can be done first for "WANT" and then reverse the palm orientation so finally palm are facing downward and using negative facial expression.

2.2.4 Conclusions

As the basic elements of the sign language are explained just above now it quite clear that the complexity of the system will be quite obvious if we consider every parameter which are

explained in morphology, phonology and syntax. As every characteristic is not feasible to implement in this project since only joints of the body parts are tracked. so we have only taken some parameters on phonology such as location (of the hand, wrist, elbow, shoulder, etc), movement and plane.

1. Location: The joints provided gives the location of various parts of the body such as hand, wrist, elbow, shoulder etc.
2. Movement: when the sign is performed the hand is moved from one location to another location, the joints also changes their position and the movement of the joints is tracked.
3. Plane: when the sign is performed, the plane changes depending on the distance of the hand from the contact to the body (the contact on the body denotes the first plane and the farthest point denotes the fourth plane).

The dataset for the different signs have been collected in appendix A.

The proposed system doesn't consider any linguistic parameters from the morphology or syntax such as word order etc. but if the system could recognize some non-manual characteristic such as the lips movements , It will be more robust. This is one of the future improvements in this project.

2.3 Sign Capturing Methods

There are many methods to capture the action performed by the user. We will discuss mainly three methods. These are (1) Data Glove approach (2) Vision-based Sign Extraction and finally (3) Microsoft Kinect based approach. The first two methods deals with 2-D data whereas the third method is based on the 3-D data.

2.3.1 Data glove method

It is one of the conventional method which employ an optical or mechanical sensor attached to the gloves which senses the hand flexions and convert that into electrical signal. These electrical signals are interpreted by the computer to determine the hand posture. This method require the glove to be worn and some tiresome device with a load of cable connected to the computer which hinders the natural human computer interaction.

2.3.2 Vision based sign extraction

This is one of the commonly used method which uses the RGB camera to get the image of the sign and creates the database for the gesture by selecting the gestures with relevant meaning. To increase the accuracy of the system multiple sample is taken for the same gesture. The extracted sample convey location, posture and motion features of the fingers, palm and face [7]. Next the signal processing is done to extract the signer hands from the background. The problem here is error associated with the environment so background removal is crucial here.

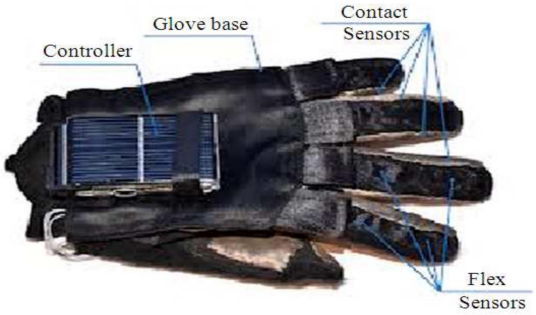


Figure 2.2: Data Glove method

Figure 2.3: Vision based method

2.3.3 Microsoft Kinect

Microsoft kinect was launched in the year 2010 as a peripheral device for Xbox 360 gaming console. Previously for gesture based controls, the proper accessories is to be wore for gaming. Now it has got three sensors, i.e. depth camera, RGB camera and microphone array. Out of the three, the most eye catching sensor was the depth camera. The depth sensor gives the depth image in which every pixel represents the distance of the object from the sensor in meters. The microphone array make it enable for controlling application with sound and RGB camera enable to identify the user. This way the players body acts as a controller in playing games. Although initially it was meant for only gaming purpose, Microsoft kinect made a way for a large number of applications in the area of computer vision such as Human Gesture Recognition, Virtual Reality and Robotics etc.

Sensors and Characteristics

Microsoft kinect has three sensors along with one motorized tilt. These are distributed over the device as shown in the figure 2.2.

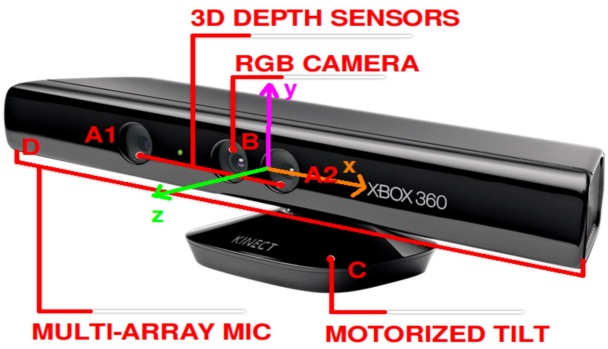


Figure 2.4: Sensors distribution of Kinect.

- *Mark A* refers to the depth sensor of the kinect. It has got two components one is CMOS sensor (A2) and other is infrared laser projector (A1). The laser projector throw the infrared beam of light in the kinect's field of view and the object reflects it back which is received by the CMOS sensor which creates a depth of map. A depth of map gives the distance of the object surface from the kinect's viewpoint and 3-dimentional scene is created out of it. This technology is developed by PrimeSense, an Israel based company. It provides the 32-bit depth with resolution 640x480 and a frame rate of 30fps. The systems creates the depth map based on time of flight i.e. the time it takes to return to the source after reflection from the object's surface in the sensor's field of view. Such types of systems are called as *Time of Flight* (ToF) systems. The optimal distance within which the user should stand or sit in sensors's view field is about 1.3 to 3.4 meters (figure 2.3 (a) and (c)).
- *Mark B* denotes the RGB camera of the kinect. It produces 2-dimentional color video of the scene. It outputs 32-bit color image with resolution of 640x 480. It has got frame rate of 30fps.
- *Mark C* is the motorized tilt which gives the tilt range the kinect can have. It gives the Kinect's field of view. The following are the specification

  – horizontal : 57.5 degrees

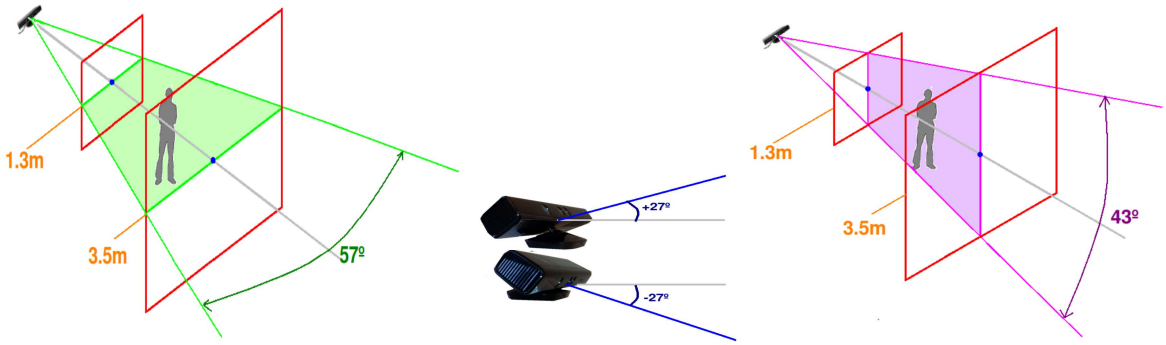
  – vertical : 43.5 degrees, with -27 to +27 degree tilt up and down side

– depth range : the physical limits 0.8 to 4m by default. This limit can be increased beyond 4m but skeleton will get noisy the further you get away so it becomes unreliable.

– The spot range (where people experience optimal interaction with having large range of movement) is 2.1m to 3.5m

- *Mark D* denotes the array of microphones in the kinect. It has got four microphones located along the bottom of the kinect. It can detect audio from +50 degree to – 50 degree in front of the it. The microphone can be pointed at 10 degree increments within 100 degrees. This feature can be used to direct towards important audio input and thus suppressing other ambient noise but it will not remove it completely. It can cancel 20db of the ambient noise which is generally whisper noise level and thus improving audio fidality. Sounds coming from behind the sensor gets 6db additional suppression thanking to the design of the microphone housing.

.



(a) Horizontal field of view (b)Tilt Range (c)Vertical field of view

Figure 2.5: Sensor's field of view

2.4 Software Architecture

To develop the Kinect based applications we need tools and APIs(Application Programming Interfaces) that is provided by the Software Development Kit (SDK) and the software used in this project was made to run on the Microsoft Kinect (SDK) v1.8. The Software development Kit contains libraries, documentations , header files, samples and tools to support and develop the applications for the Gesture and speech controlling devices such as the Kinect xbox. The overview of the kinect software layer is given in figure 2.6, which shows lowest level hardware in which four blocks are given for respective hardware that is RGB camera, depth sensor,tilt motor and Audio mic array. The mid layer consists of windows library which contains drivers for natural user interface and Application processing interface and libraries for audio sensors etc.The third layer gives the application which are build on these software for kinect such as gesture to speech convertor, speech to sign converstor etc.

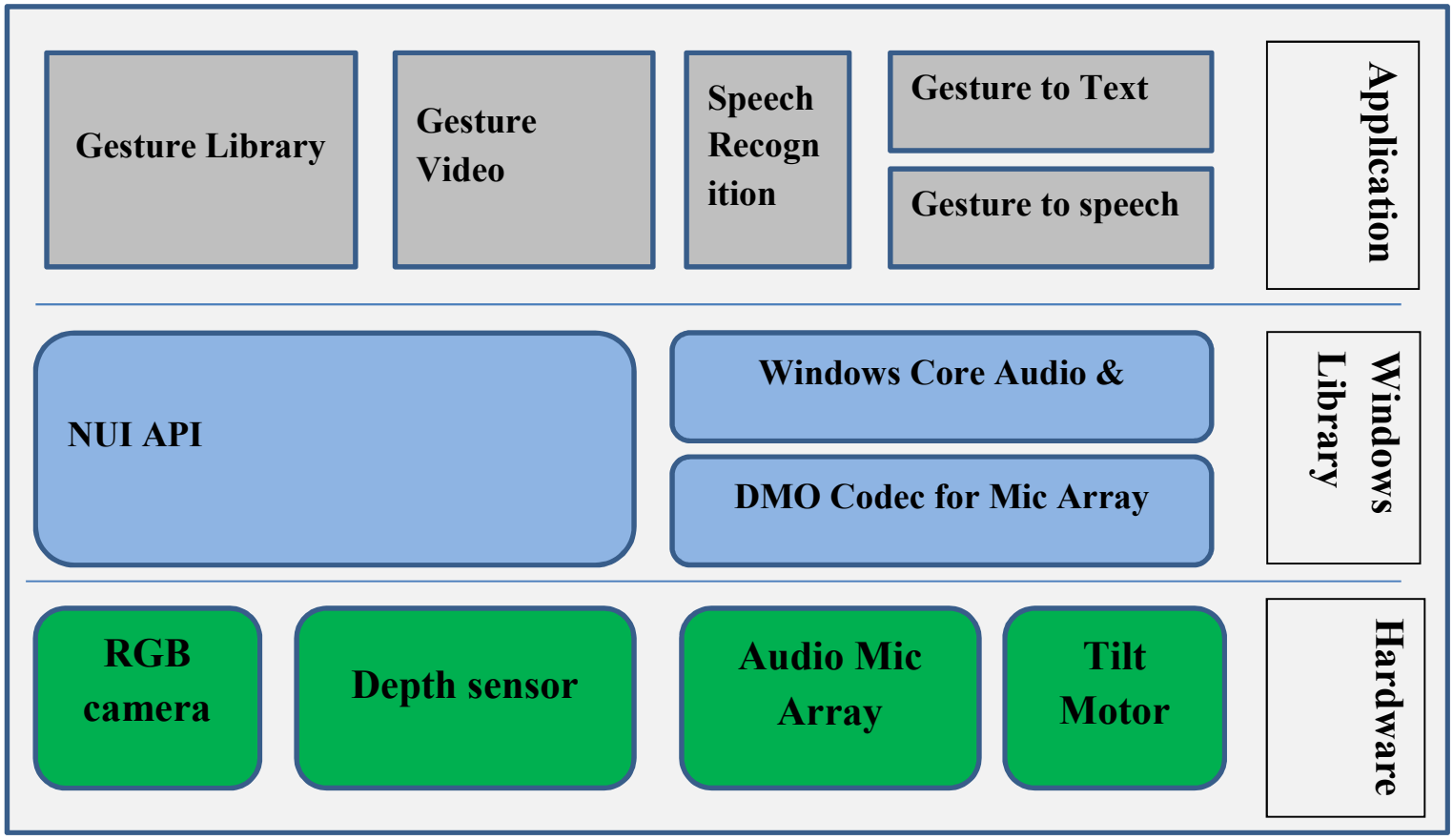


Figure:2.6 Overview of Kinect software layer

The following hardware is required for this project

1. Kinect XBOX 360, including the Kinect sensor and the USB hub, through which the sensor is connected to the computer
2. 64-bit (x64) processors
3. Dual-core, 2-GHz or faster processor
4. USB 2.0 bus dedicated to the Kinect
5. 2 GB of RAM (4 GB recommended)

The software requirements are

1. Kinect SDKver.1.8 for the Kinect sensor.
2. Windows 7 standard APIs- The audio, speech, and media APIs in Windows 7, as described in the Windows 7 SDK and the Microsoft Speech SDK.
3. Microsoft Visual Studio 2010 or higher version.
4. .NET Framework 4 (installed with Visual Studio 2010)

To enable Kinect based application for windows platform, it makes used of Software Development Kit (SDK) framework, which works in many layers. This is very important to have a basic understanding of various layers and their interactions. At the lowest level, the kinect SDK provides the requisite drivers which generates the output from the sensors such as visual data (from depth and RGB sensor) as well as audio data. The depth data along with skeleton tracking of the corresponding RGB image can be seen in figure 2.6.

Controlling of the streaming audio and video (depth, color and skeleton) provided by the kinect sensor is done by the drivers installed as a part of SDK.

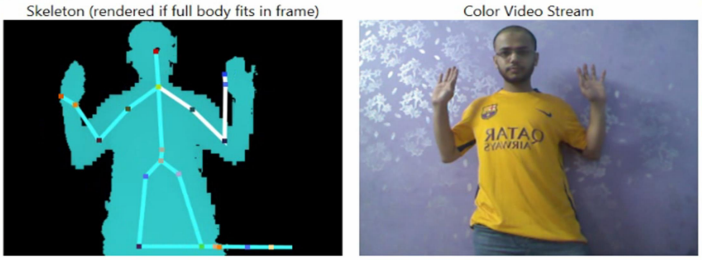


Figure 2.6 : Depth data with Skeleton

Chapter 3

3 Methodology

This section will deal with the explanation of procedures and moves taken during the development of this project. There are two independent modules in this system. The first module is gesture/sign to speech/text conversion and second module is speech to sign/gesture conversion. First of all, a general overview of both of modules are explained then each block in that module is explained.

3.1 Gesture to Speech and Text Conversion Module

This module of Gesture to Speech/Text conversion converts the sign or gesture performed by the signer by placing itself in sensor's field of view into speech/text for the corresponding sign. The block diagram for this module is shown in Figure 3.1. When the user perform some sign or tries to do so in kinect's field of view, the new frames are acquired and video stream gets updated with the joints of the skeleton of the user overlapped onto it. At this instant, if the user wishes to record the frames (otherwise camera will take new frames to the system), the following blocks are executed sequencially i.e. Obtain joint of interest block, normalize data block and built frame descriptor block. In the first block, the joints of interest are obtained for describing the frame descriptor, normalization of the joints are done in second block and finally frame descriptor is defined for each sign in the dictionary.

There are two mode of working i.e. Training (capture) mode and Testing mode. In Capture (training) mode, the user can add new signs to the dictionary and frame descriptor is added to the dictionary while in Testing mode, the translation is done of the sign being performed and frame descriptor is added to a file "RecodedGesture". Once the sign is finished, which the system know if the consecutive frames occur multiple times, then based on the working mode, if the mode is TESTING then the test gesture is compared using the classifier with the signs stored in the gesture dictionary and the corresponding output is displayed in text as

well as in spoken word, so that user understands the sign. After that system waits for the next frame and the flow within the block diagram repeats again.

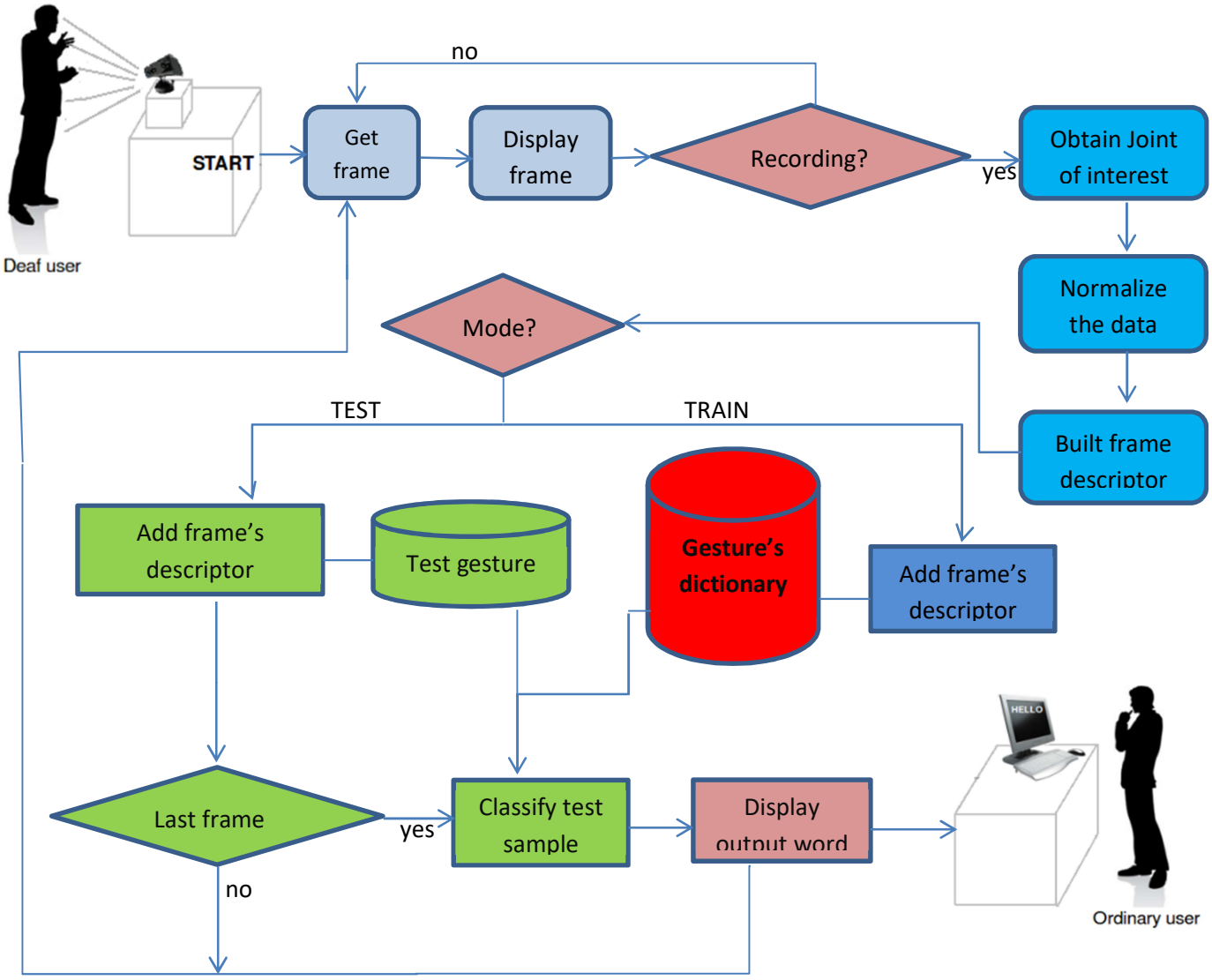


Figure 3.1: Block diagram for the system of Sign to Text conversion.

The following sections analyse and explain the different blocks of the system in more detail.

3.2 Joints of Interest (JoI)

Microsoft visual studio along with software development kit provides 20 joints of the human body i.e. of head, neck, right shoulder, right elbow, right wrist, right hand, left shoulder, left elbow, left wrist, left hand, hip center, torso, left hip, right hip, left tigh, right tigh, left ankle,

right ankle, left foot and right foot. But for sign language purpose these 20 joints are too much we require only 4 joints i.e. right elbow, left elbow, right wrist, left wrist. These are only significant for the description task of the sign. So there is no point in tracking other joints such as neck, shoulder, foot etc. of the human body since it remain static during signing and its execution. If we add these joints also, these joints will work as redundant only.

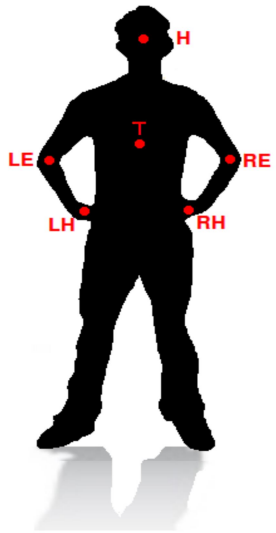


Figure 3.2 Used joints

For signing description only four mentioned joints are used but for normalization purpose we require two more joints i.e. head and torso. By doing so the total list of joints required now reduce to six which are indicated in figure 3.2.

3.3 Data Normalization

The motto behind the normalization of the data is to make the system robust for size invariance and position invariance. The user can be of different heights so our system should robust to take care for that and the user can be in any position within the kinect's field of view so our system should be capable to produce correct output in either case. Hence, the normalization of joints becomes one of the most crucial task in order to achieve the goal of size and position invariance. Hereafter the details of how to deal with both will be explained.

3.3.1 User's position invariant

The user can place itself anywhere in the room and accordingly the coordinates of the joints will be stored relative to its position. Normalization will consider the position of the user.

As shown in figure 3.3, the user position varying in the room drastically changes the value of joint's coordinate. A little variation in depth can cause huge changes in the values of X, Y and Z coordinates.

The proposal of normalization of the joints coordinates is done with respect of the position of torso, instead of storing values directly in Cartesian coordinates X,Y and Z. Since the position of torso remains almost constant along the frame it would be the right choice to choose this joint, to make system position invariant. The position of torso is found by averaging the coordinates of the left shoulder and right shoulder.

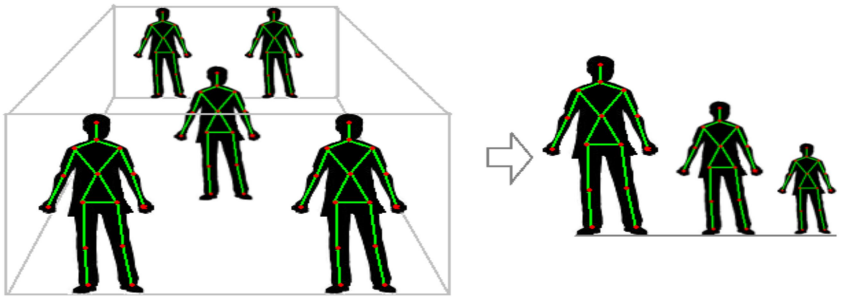


Figure 3.3: User at different position of the room

Let the set of joints J={LH,RH,LE,RE} and T as a position of torso.

Position of Torso, T(T_x, T_y, T_z) where T_x and T_y is given by $T_x = \frac{LS_x + RS_x}{2}$, $T_y = \frac{LS_y + RS_y}{2}$ and $T_z = \frac{LS_z + RS_z}{2}$

Where LS=left shoulder and RS=right shoulder.

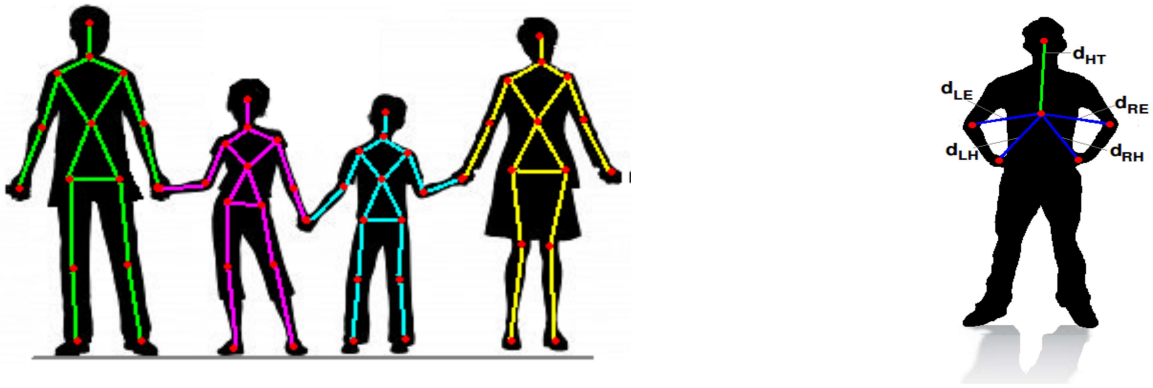
Consider the sets of distance D = $\{d_{LH}, d_{RH}, d_{LE}, d_{RE}\}$, centering of data joints is done as follows

$$\sum_{i=1}^{n} D(i) = \sqrt{(J(i)_x - T_x)^2 + (J(i)_y - T_y)^2 + (J(i)_z - T_z)^2}$$

Where n denotes the number of joints in J [15].

3.3.2 User's size invariant

For a given sign, the frame descriptor should be same no matter if the user is short or tall and the system should produce the same output in either case. Figure 3.4 shows the user of different heights and sets of distances D.



(a) Users of varying heights (b) set of distances D

Figure3.4 : Normalization for user's height

We have chosen distance d_{HT} between torso and head since this distance will vary depending on the size of the user i.e. if the user is tall then this distance will be lagre and if user is of short height then this distance will be smaller.

If we divide every other distance in set D = $\{d_{LH}, d_{RH}, d_{LE}, d_{RE}\}$ then we will obtain normalized distance i.e. D_{norm}.

$$\sum_{i=1}^{n} D_{norm} = \frac{D(i)}{d_{HT}}$$

Where n is the number of distances in D and d_{HT} is the distance between head and torso (green segment in the image above (b)).

The positive effects of these normalizations can be seen in results section.

3.4 Gesture descriptor

After obtaining JoI and normalization of it, the next step is the building the descriptor for each gesture/sign. No two signs can have same descriptor. Each descriptor must be unique and different from other descriptor in the dictionary.

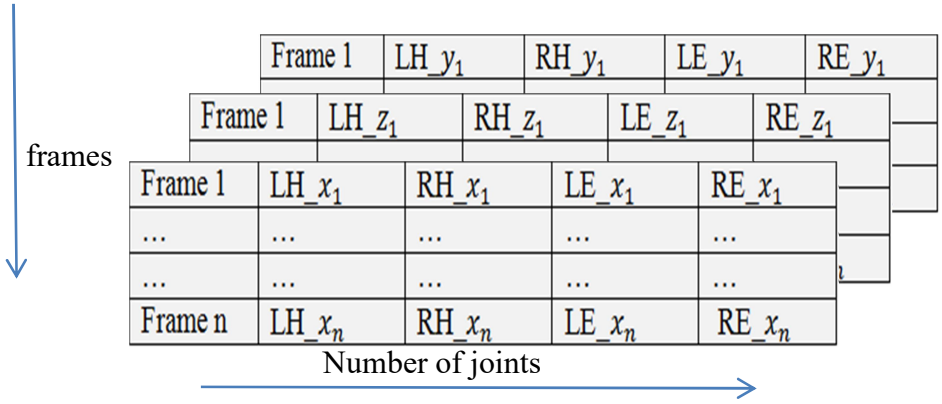


Figure 3.5: sign descriptor

The each descriptor contains as many rows as number of frames captured for one gesture and as many columns as the number of joints. The number of layers for each Cartesian coordinates.

3.5 Classifier

The classifier is a function which classify the input gesture to the correct output. While choosing the classifier the following points should be taken into account.

- the classifier should be able to compare data of varying length, sometimes the same sign can be performed by same user may have different length.
- The classifier should compare every test sign with the signs stored in dictionary.

The section 3.5.1 gives description about Dynamic Time Warping (DTW) algorithm and section 3.5.2 deals with the classifier used i.e. Nearest Neighbor along with Dynamic Time Warping algorithm as a cost function.

3.5.1 Dynamic time warping

The dynamic time warping has been used excessively in many fields since its inception in 1960, such as in speech recognition, hand writing recognition, online signature matching etc. and now in gesture recognition.

This algorithm finds the similarity between two time series which are varying in time and speed so can be applied to any time varying series. DTW is widely used in speech recognition to find whether two waveforms represent the same spoken word or not. We have used this algorithm to find whether the sign performed by the user matches with the sign stored in the dictionary since sign or gesture being dynamic is a time series which vary in time as well as speed.

Theoretical background

There are many types of learning algorithm. The main two types are supervised and unsupervised learning algorithm. In supervised algorithm, the users are going to teach the machine to do something, whereas in unsupervised learning algorithm, user says let the machine learn by itself. Supervised learning refers to the fact that the algorithm contains a data set in which correct output are given.

The DTW used here is an example of supervised learning and it is basically an classification problem. Classification means prediction which outputs a discrete value.

For example, we have a gesture dataset consisting of 2 dimentional axis position of the joints for specific sign. Given a dataset like this the classification algorithm can draw straight line between dataset to separate the valid sign from zero signs.

The goal of DTW is comparison between two time dependent series X=$(x_1, x_2, \dots, x_n)$ of length N ∈ **N** and Y=$(y_1, y_2, \dots, y_n)$ of length M ∈ **N.** these time dependent series may be feature sequences which are sampled at equidistance points in time. Let F denotes the feature space set such that

$x_n, y_m \in \mathbf{N}$ for n ∈ [1,N] and m ∈ [1,M]

To compare two different features x,y ∈ F, a local cost measure is required, sometimes also referred to as local distance measure, which is defined to be function

$$c : F \times F \rightarrow \mathbf{R} \geq \mathbf{0}$$

if x and y are similar to each other then c(x,y) is small (showing low cost) and if the similarity between x and y increases then c(x,y) also increases.

Algorithm starts by building a cost matrix C ∈ $\mathbf{R}^{N \times M}$ representing all the pairwise distance between X and Y, see figure 3.7 [14].

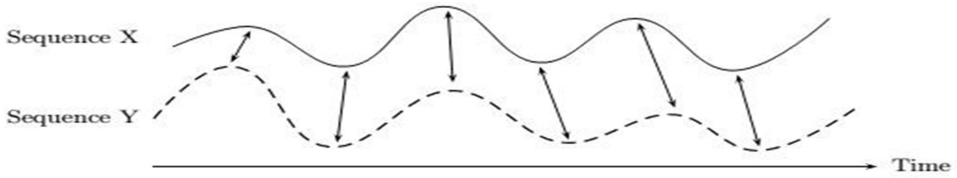


Figure 3.6: Raw time series, arrows show the desirable points of alignment.

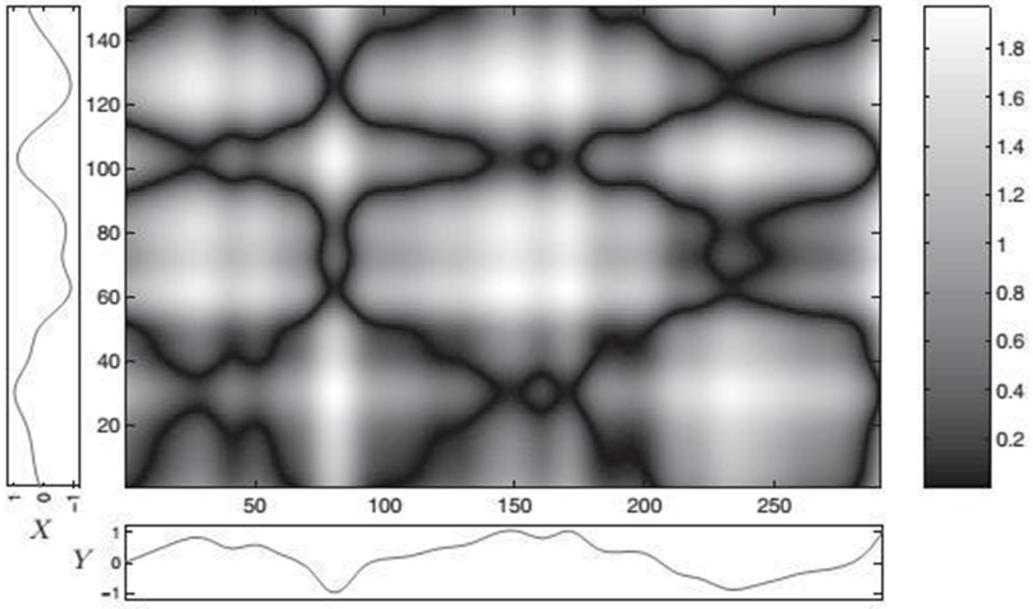


Figure 3.7: The optimal warping path aligning two time series.

The local cost matrix for the alignment of the two real valued sequences X and Y using Manhattan distance as local cost measure C is given by

$$C \in \mathbf{R}^{N\times M}: C_{i,j} = \|x_i - y_j\|,\ \text{i}\in[1{:}\text{N}], \text{j}\in[1{:}\text{N}]$$

When the local cost matrix created, the algorithm finds the alignment path that runs through the low-cost areas “valleys" on the cost matrix. This warping path defines the correspondence of an element $x_i \in X$ to $y_i \in Y$ following the boundary condition that assign first and last elements of X and Y to each other (Müller, 2007).

The warping path built by Dynamic Time Warping is a sequence of points $p = (p_1, p_2.., p_l)$ with $p_l = (n_l, m_l) \in [1{:}N] \times [1{:}N]$ for $l \in [1{:}N]$ must satisfy the following criteria:

1. **Condition of Monotonicity:** $n_1 \leq n_2 \leq \cdots \leq n_L$ and $m_1 \leq m_2 \leq \cdots \leq m_L$. This condition prevents the time ordering of the points.
2. **Step size condition:** $p_{l+1} - p_l \in \{(1,0), (0,1), (1,1)\}$ for $l \in [1{:}L-1]$. This condition prevents the warping path from having long jumps i.e. shifts in time while aligning sequences.
3. **Boundary condition:** $p_1 = (1,1)$ and $p_L = (N, M)$. The starting and end points of the warping path must be the first and last points of the aligned sequence.

Although the introduced DTW algorithm works with 1-dimensaional data. So we have modified it to use for our 12-dimentional frame descriptor. In our case, we have sequence of the frame descriptor instead of time series as equation below

$$[LH_{x_1}, LH_{y_1}, LH_{z_1}, RH_{x_1}, RH_{y_1}, RH_{z_1}, LE_{x_1}, LE_{y_1}, LE_{z_1}, RE_{x_1}, RE_{y_1}, RE_{z_1}]$$

Above frame descriptor has 12-dimentions which can be reduced to 8-dimensions if we neglect the z-component. Hereafter is shown the way to modify the DTW algorithm for two sequences of n-dimensional data.

The modification exists at a point that the cost between the two n-dimensional data is computed. Given a n-dimensional data X1 and X2 , the cost value between them can be generalized as follows:

$$\text{Cost}=\sqrt{\sum_{i=1}^{n}(X1_i - X2_i)^2}$$

3.5.2 NN-DTW classifier

The Nearest Neighbor Dynamic Time Warping classifier (NN-DTW) is a modified version of well known Nearest Neighbor classifier along with Dynamic Time Warping Algorithm as a cost function. For given test sign, the classifier finds the best match sign from the dictionary. In order to find the similarity between the given sign and any sign from the dictionary, DTW algorithm is used.

Here is the pseudo code for this classifier.

Algorithm 3.1: Nearest Neighbor Classifier using Dynamic Time Warping cost function

---

```
function NN-DTW (sample test, vector < sample > dictionary [1..n])
    declare double  min,dist
      for i := 1 to n do
        dist ← DTW distance (test; dictionary[i])
          if  dist ≤min  then
            min   ← dist
          end if
      end for
  return return min
end function
```

3.6 Speech to Gesture Conversion Module

The system of ours also includes a module that enable speech recognition of commands and gives the developer for the library in Kinect SDK. The Kinect sensor captures both voice and gesture including face tracking and gesture from small to whole body. The sensor equipped with four microphones that enable applications to respond to oral input, besides responding to gesture. The sensor detects audio input in range 100 degrees in front of the sensor. The microphone arrays can be placed at 10-degree increments within the 100-degree range as shown in Figure 3.8. This could be exploited to specifics about the direction of the most important voices, like people talk, however it will not entirely eliminate noise in from environment. Kinect is programmed to follow the loudest audio input. Microphone arrays may cancel 20 dB (decibels) of the ambient noise which is equivalent to noise levels of whisper [2].

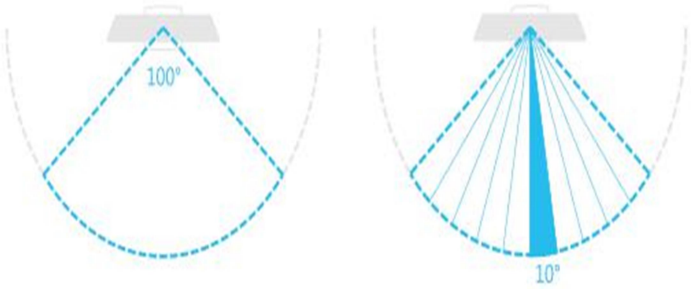


Figure 3.8: Kinect audio input range.

User may choose a specific words or phrases to command the kinect sensor. Using conversation as command is not recommended since sensor with treat it as noise. Even it might not look as a natural interaction but voice input is designed and integrated well, kinect responds faster and makes the experience better. Some of the phrases that often used as command are “One”, ”Two”, ”Left”, ”Right”, ”Happy”, ”How are you” etc.

3.6.1 Implementation approach

The strategy of implementation of Speech to Gesture Module is illustrated in figure 3.9.The system will get the speech input from the user that will be processed and get converted into related text form which is done with the help of external library for speech to text conversion which takes word or sentence as input, process it and identify the keyword in the input.

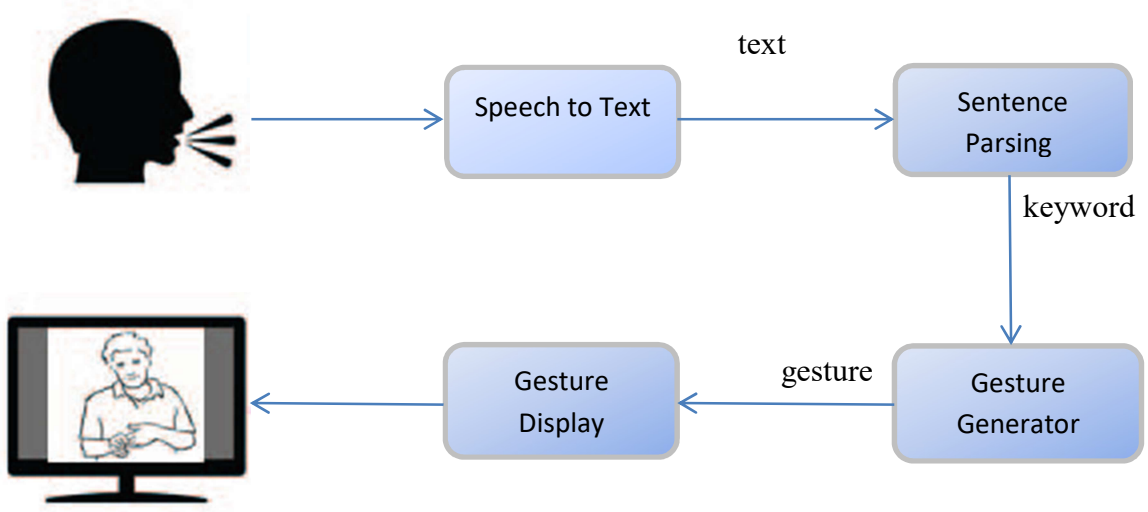


Figure 3.9: Speech to sign module workflow

Once the keyword is known, the next step is to search for sign/gesture against the keyword from the database. After getting the gesture to the corresponding keyword, it is transferred to the video display terminal. In the video display terminal the corresponding gesture is shown. This gesture can be seen by the deaf and can interpret it.

Chapter 4

4 Experiments and results

This chapter discusses the evaluation of Kinect Sign Language user experience and testing with the experiments performed and finally accuracy of the system for both the modules are analysed. User experience refers to all the facets of one's interaction with the system.

4.1 Setting the environment`

The environment and surrounding have a significant influence on the sensor and hence on system performance. A proper environment setup is required to overcome these influencing factors before running the test. These limitations have been controlled by following the guidelines from Kinect Human Interface Guideline v1.8(2013). These guidelines are as follows

1. Distance between user and kinect

The Microsoft kinect can track skeleton of two persons completely but according to the guidelines , the number of users are restricted to one. A boundary is marked on the floor to give a hint, the area of optimum interaction of the user with the device, see figure 4.1. The 3D image of the environment setting is made using software SketchUp Make.

Figure 4.1: Kinect environment setting.

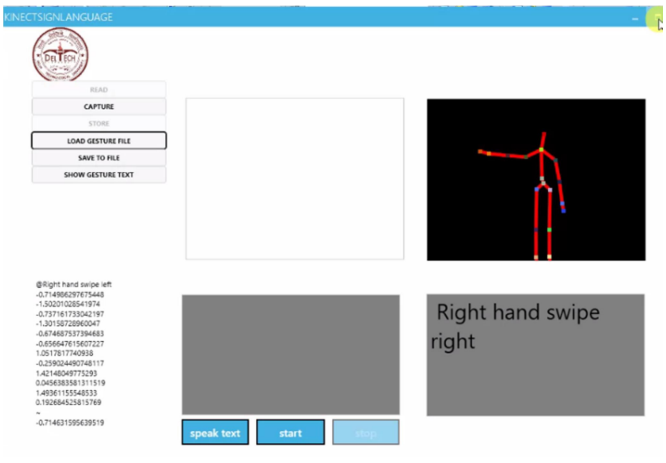
2. Voice input

To avoid mixing of voice input with other voices , the system should be placed in such an environment where minimum noise is there. This will make sure of good capturing of the voice input by the device.

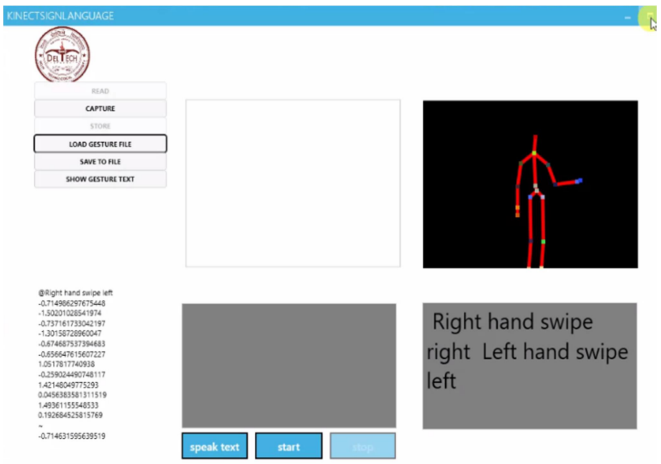
After setting the environment for the experiment, the next step is to evaluate the system. We have 10 signs in the dictionary, listed in Table1.1 which is recognized by the first module of sign/gesture to speech and text conversion and 14 spoken words in the dictionary given in Table 1.2 for the evaluation of the second module of speech to sign conversion. We have chosen three testers to test the accuracy of both the system. These testers are of different heights and body shapes.

4.2 Experiments with the first module

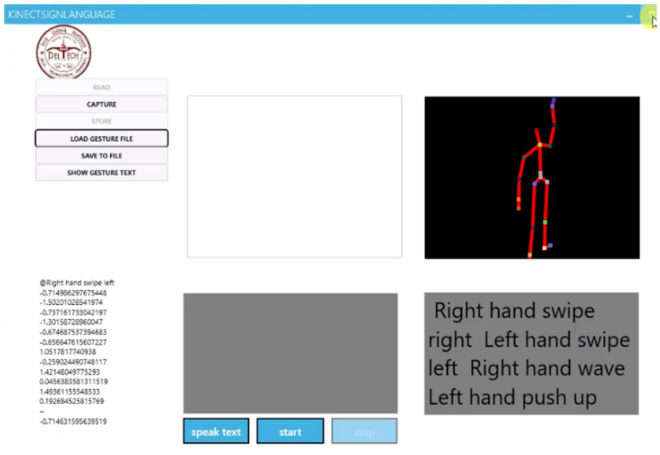
In the first module of the system (sign to speech/text), all the three testers perform sign for each word in the dictionary, see Table1.1 ten times. Each sign is performed at different speed which is taken care by the Dynamic Time Warping Algorithm to evaluate the system. Some of the signs performed and results are shown in the figure 4.2. The signs which are recognized and are classified correctly or misclassified are shown in the output box and speech for the word is also generated and when sign is not recognized by the system it remains silent.



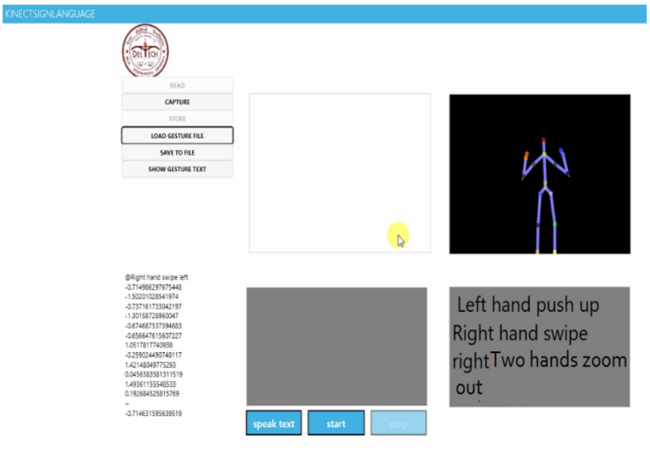
(a) Sign for "Right hand swipe right"



(b) Sign for "Left hand swipe left"



(c) Sign for "Left hand push up"



(d) Sign for "Two hands zoom out"

Figure 4.2: experiments for sign to speech/text module

The results of the experiments are listed in the Table 4.1. All the sentences in the dictionary are put along the rows of the table and testing results along the column of the table. For example: the sign for "Right hand swipe right" is performed by first tester ten times and system classified the sentence correctly eight times. Similarly tester 2 and tester 3 classified the sentence correctly nine times each. The accuracy of the system is found by applying the formula listed below

$$Accuracy = \frac{\sum_{i=1}^{n} \frac{positives_i}{positives_i + negatives_i} \cdot 100}{k} \quad \ldots(1)$$

Where $positives$ are the test samples which are rightly classified and $negatives$ are the ones that are classified wrongly. k denotes the number of testers involved (three here) and n denotes the number of signs which are 10 in our case.

Using the above formula, the accuracy of each sign is calculated and are listed in the last column of the table and Averaging is done to calculate the accuracy of the system which is listed at the bottom of the table.

Sign	Tester 1	Tester 2	Tester 3	**Acuuracy (%)**
Right Hand Swipe Right	8/10	9/10	9/10	86.665
Left Hand Swipe Right	9/10	10/10	9/10	93.324
Right Hand Wave	8/10	8/10	9/10	83.325
Left Hand Wave	8/10	9/10	8/10	83.325
Right Hand Swipe Left	9/10	9/10	8/10	86.665
Left Hand Swipe left	8/10	9/10	9/10	86.665
Right Hand Push Up	9/10	9/10	10/10	93.324
Left Hand Push Up	9/10	10/10	9/10	93.324
Two Hands Zoom out	7/10	8/10	9/10	79.992
Both Hands Push up	8/10	8/10	9/10	83.325
Accuracy of the system(%)	**83%**	**89%**	**89%**	**87%**

Table 4.1: Results obtained for sign to text module

The accuracy of the overall system can also be found by finding the accuracy of system by each tester and averaging them. The accuracy of system for each tester can be found by

$$Accuracy = \frac{\sum_{i=1}^{n} \frac{positives_i}{positives_i + negatives_i} . 100}{n} \quad \ldots(2)$$

Where $positives$ are the test samples which are rightly classified and $negatives$ are the ones that are classified wrongly. k denotes the number of testers involved (three here) and n denotes the number of signs which are 10 in our case.

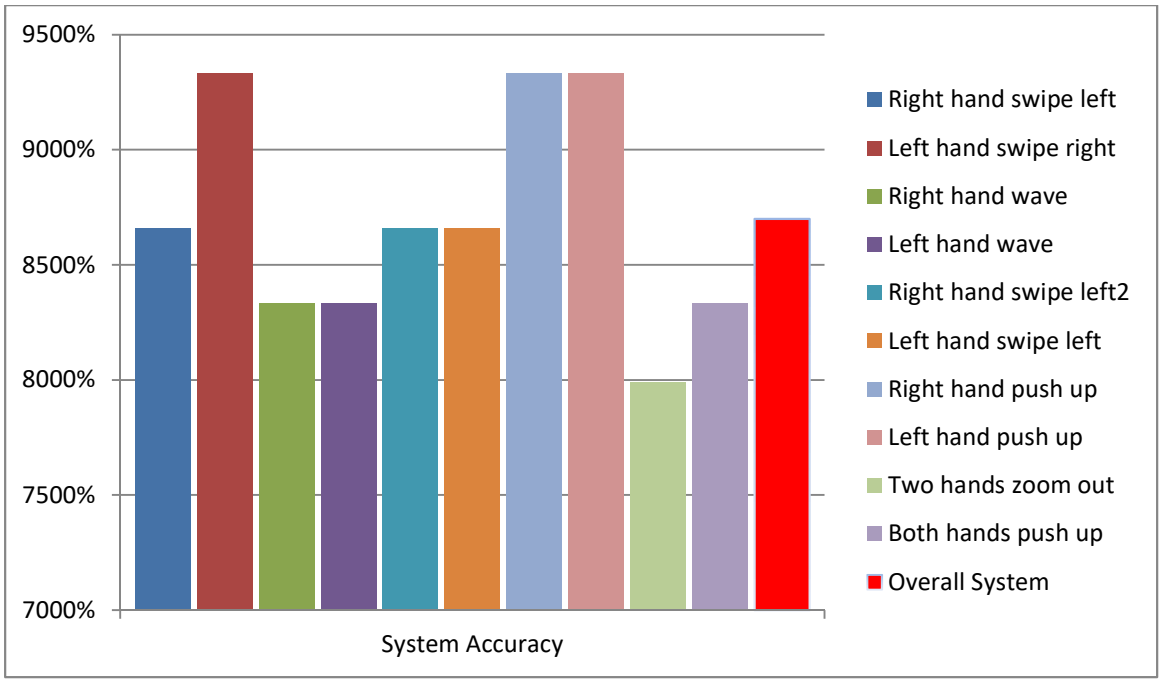


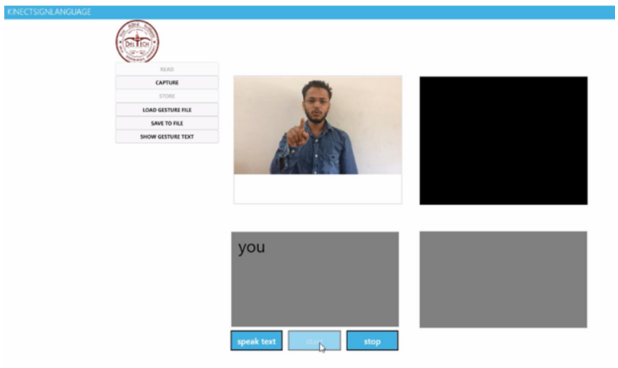
Figure 4.3 System accuracy for the different signs

We have collected 300 samples for the test of the system.Out of 300 samples taken from three different testers, only 39 samples are misclassified and 261 are classified correctly by the system. Tester 1 found the system 83% accurate while Tester 2 and Tester 3 found it to be 89% accurate.

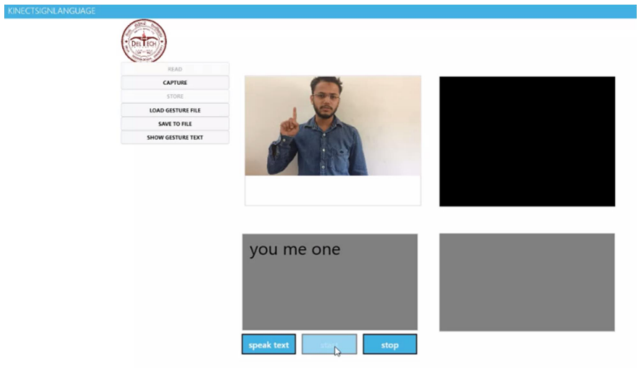
So the overall system is 87% accurate.

4.3 Experiments with the second module

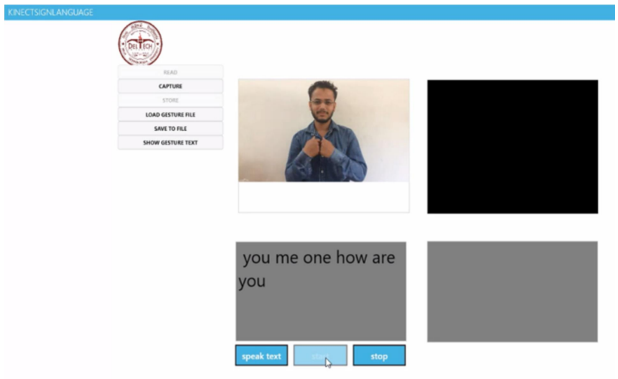
In the second module of the system (speech to gesture), all the three testers speak each word in the dictionary, see Table1.2 ten times with pause till the corresponding gesture is shown on the screen. Some of the words are spoken and corresponding gestures are shown in the figure 4.3. The words which are recognized and are classified correctly or misclassified are shown in the output box as a gesture and words which are not recognized by the system it remains silent till user speaks again.



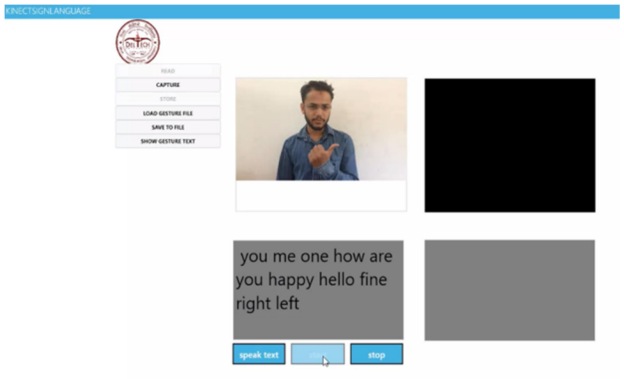
(a) Output gesture for word ”you



(b) output gesture for word ”one”



(c) output gesture for ”how are you”



(d) output gesture for “left”

Figure 4.4: experiments for speech to gesture module

The results of the experiments are listed in the Table 4.2. All the words which the system can recognized are put along the rows of the table and testing results along the column of the table. For example: the word ”Hello”is spoken by first tester ten times and system produced correct gesture nine times. Similarly tester 2 and tester 3 spoke same word “Hello” and system produced correct gesture ten times. The accuracy of the system is found by applying the formula listed in equation (1) and (2).

words	**Tester1**	**Tester2**	**Tester3**	Accuracy(%)
Hello	9/10	10/10	10/10	96.657%
Happy	10/10	9/10	10/10	96.657%
Goodbye	9/10	9/10	10/10	93.324%
Good Morning	10/10	9/10	9/10	93.324%
How are you	8/10	9/10	9/10	86.658%
Left	10/10	9/10	10/10	96.657%
right	9/10	9/10	10/10	93.324%
Four	8/10	9/10	9/10	86.658%
Three	8/10	8/10	9/10	83.325%
me	9/10	9/10	9/10	89.991%
Five	9/10	8/10	9/10	86.658%
Accuracy of the system(%)	**90%**	**89.1%**	**94.5%**	**91.2%**

Table 4.2: Results obtained for speech to gesture module

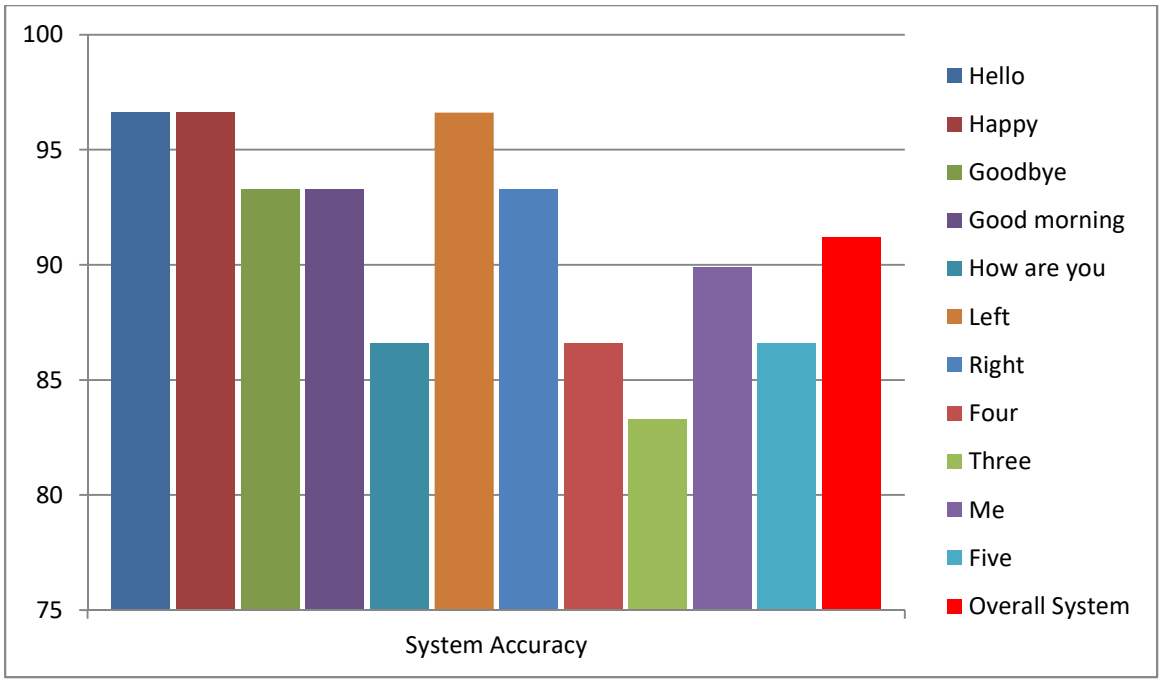


Figure 4.5: System accuracy for different spoken words

For this module, total 330 test samples were used to test the system by three different users and the system classified 301 test samples correctly and only 29 samples are wrongly classified.

After applying the formula, the tester 1 found the system to be 90% accurate and tester 2 found it 89% accurate while tester 3 found it be 94.5% accurate. so averaging of the accuracy give us the overall accuracy of the system i.e. 91.2%.

Chapter 5

5 Conclusion

The thesis proposal clearly summarized the aim of the project: to design and implement a system capable of recognizing a list of 10 sings which is the first module of the system (sign to speech/text) and understanding the 14 basic words and translating them into corresponding gesture which is the second module (speech to sign).

In the first module, for the 10 signs stored in the dictionary, the only data used is the joint positions of the hand. So it becomes very crucial to select the signs considering the data that is used to describe them. For example: the proposed system will not be able to discriminate between the two signs if they are performed in the same way and they are different in only hand shape or position of the fingers.

The first module of the system with best configuration gives the accuracy of about 87%. For this module, 300 test samples for all the 10 signs in the dictionary were collected by the three different users and applies to the system, the system misclassified only 39 samples and 261 samples are classified correctly. The second module of the system provides accuracy of 91.2%. For this module, total 330 test samples were used to test the system by three different users and the system classified 301 test samples correctly and only 29 samples are wrongly classified.

Despite the fact that the given signs do not belong to a specific official sign language, the idea of the project was to show that with basic descriptors and classifiers and the use of the Kinect, a wide number of signs could be recognized and the system has the potential to provide a computationally efficient design without sacrificing the recognition accuracy of the system.

Such a system with more than 90% accuracy can be used in public places such as at airport, railway and hospitals to reduce the communication barrier between deaf and mute people in the society.

Other secondary goals have been also satisfied. While using Microsoft visual studio for C# as a programming language, the knowledge about the language along with .NET Framework is learned. The SDK installation along with its drivers interfaced with visual studio improves the knowledge about the set up an external device and its working.

5.1 Future work

To make this system work with a real Sign Language (American Sign Language, Spanish Sign Language, etc.), some other features such as the finger position or shape of the hand will have to be considered. This would probably be the most appropriate first future improvement. A sign is not just a combination of hand gestures, other components come into play, with the lip movement possibly being one of the most important. Considering these other components will allow the project to develop into a more true Sign Language Translator.

Another future development will be to implement a dynamic algorithm to detect whether the user is doing something that seems to be a sign from the dictionary. Right now, the user has to start a sign with a specific pose so that the record of a test will start. It will be interesting to make the gesture recognition task more automatic without the need of an initial position.

Bibliography

[1] Mateen Ahmed, Mujtaba Idrees, Zain ul Abideen, Rafia Mumtaz, Sana Khalique "Deaf talk using 3D animated sign language: A sign language interpreter using Microsoft's kinect v2",IEEE(2016). SAI Computing Conference 2016 July 13-15, 2016 | London, UK

[2] *Kinect for Windows | Human Interface Guideline v1.8.* (2013). Microsoft Corporation.

[3]Zafar Ahmed Ansari and Gaurav Harit 2013 Nearest Neighbour classification of Indian Sign Language gestures using Kinect camera.In:Indian Academy of Science,Sadhana Vol.41,No.2,February 2016,pp.161-182.

[4] V.I. Pavlovic, R. Sharma, and T.S. Huang. Visual interpretation of hand gestures for human-computer interaction: a review. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 19(7):677 {695, jul 1997.

[5] D. M. Gavrila. The visual analysis of human movement: A survey. Computer Vision and Image Understanding, 73:82{98, 1999.

[6] Ying Wu and Thomas S. Huang. Vision-based gesture recognition: A review. In Proceedings of the International Gesture Workshop on Gesture- Based Communication in Human-Computer Interaction, GW '99, pages 103{ 115, London, UK, UK, 1999. Springer-Verlag.

[7] Sutarman, Mazlina Abdul Majid and Jasni Mohamad Zain. A Review on the development of Indonesian sign language recognition system, Journal of Computer Science 9 (11): 1496-1505, 2013 ISSN: 1549-3636. doi:10.3844/jcssp.2013.1496.1505 Published Online 9 (11) 2013 (http://www.thescipub.com/jcs.toc)

[8] Yang Quan and Peng Jinye. Application of improved sign language recognition and synthesis technology in ib. In Industrial Electronics and Applications, 2008. ICIEA 2008. 3rd IEEE Conference on, pages 1629 {1634, june 2008.

[9] Dai Jun. Gesture recognition reach based on high-order nmi. Master Dissertation, ShanHai Maritime University, May 2004.

[10] Yang Quan. Sign language letter recognition algorithm based on 7hu invariant moments. Master Dissertation, Xi'ans University of Architecture and Technology, July 2007. [11] R. Akmeliawati, M.P.-L. Ooi, and Ye Chow Kuang. Real-time Malaysian sign language translation

using colour segmentation and neural network. In Instrumentation and Measurement Technology Conference Proceedings, 2007. IMTC 2007. IEEE, pages 1 {6, may 2007}.

[11] Jonathan C. Hall. Gesture recognition with kinect using hidden markov models (hmms). http://www.creativedistraction.com/demos/gesture-recognition-kinect-with-hidden-markov models-hmms/.

[12] Christian Vogler and Dimitris Metaxas. A framework for recognizing the simultaneous aspects of american sign language. Computer Vision and Image Understanding, 81:358{384, 2001.

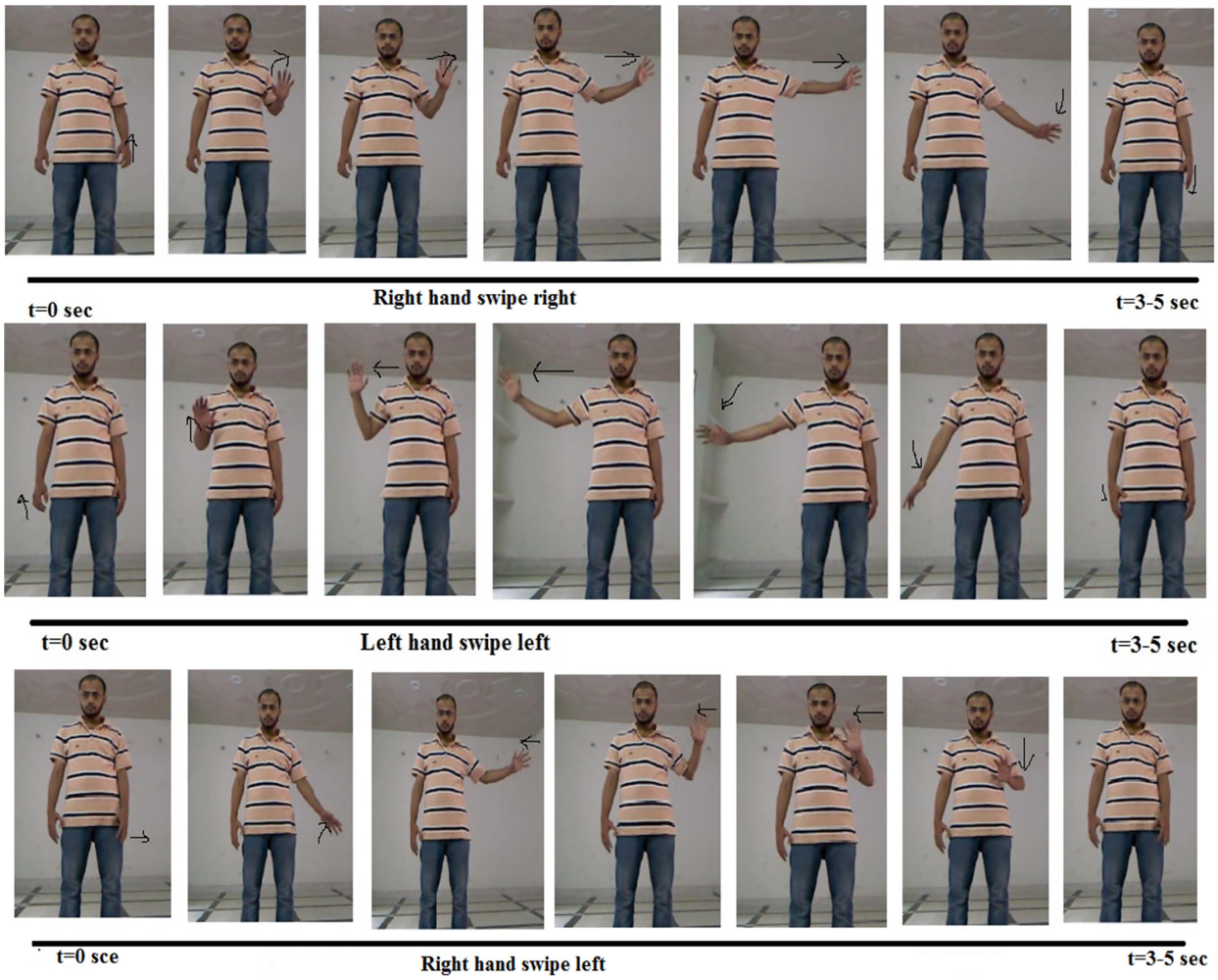
[13] Thad Starner, Joshua Weaver, and Alex Pentland. Real-time american sign language recognition using desk and wearable computer based video. IEEE Transactions on Pattern Analysis and Machine Intelligence, 20:1371{1375, 1998}.

[14] Gibran García-Bautista, Felipe Trujillo-Romero, Santiago Omar Caballero-Morales. "Mexican Sign Language Recognition Using Kinect and Data Time Warping Algorithm" Universidad Tecnológica de la Mixteca.

[15] J. Xu, "Sign Language Translation Using Kinect And Dynamic Time Warping" http://web.stanford.edu/class/cs231m/projects/final-report-xu.pdf/, (17 July 2016).

6 Appendix A

In this appendix, we have added how to perform different signs for the Sign to Speech Translator module. These signs are added by one tester just to tell how the different sign can be performed to make the system more accurate and fast.



t=o sec

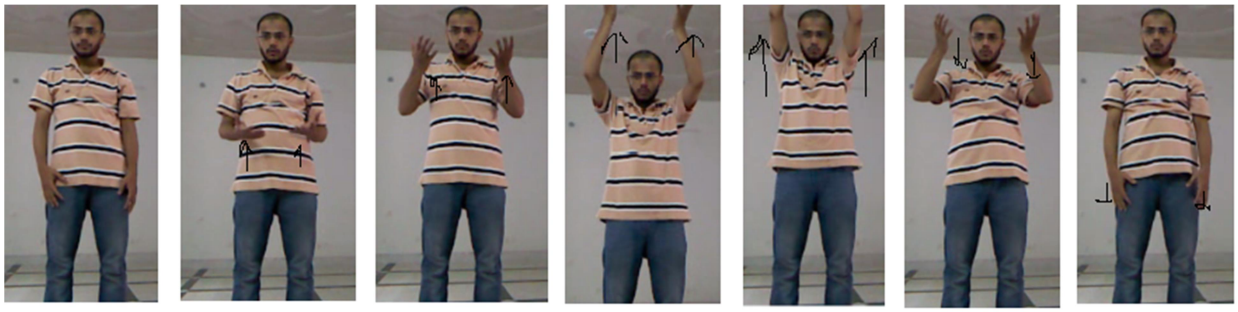
Left hand swipe right

t= 3-4 sec

t=0 sec

Left hand push up

t=3-5 sec



t=0 sec

Both hand push up

t=3-5 sec

t=0 sec

Two hands zoom out

t=3-5 sec

t=0 sec Right hand wave t=3-5 sec