# Natural User Interface using Microsoft Kinect and HTK

*Thesis submitted in partial fulfillment of the requirements for the award of degree of*

**Master of Engineering**

in

**Information Security**

*Submitted By*

**Umesh Kumar**

**(Roll No. 801333029)**

Under the supervision of

**Dr. Parteek Kumar**

Assistant Professor

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT

THAPAR UNIVERSITY

PATIALA – 147004

**July 2015**

# CERTIFICATE

I hereby certify that the work which is being presented in the thesis entitled, "*Natural User Interface using Microsoft Kinect and HTK*", in partial fulfillment of the requirements for the award of degree of Master of Engineering in *Information Security* submitted in Computer Science and Engineering Department of Thapar University, Patiala; is an authentic record of my own work carried out under the supervision of Dr. Parteek Kumar and refers other researcher's work which are duly listed in the reference section.

The matter presented in the thesis has not been submitted for award of any other degree of this or any other University.

**(Umesh Kumar)**

This is to certify that the above statement made by the candidate is correct and true to the best of my knowledge.

**(Dr. Parteek Kumar)**
Assistant Professor
Thapar University
Patiala

**Countersigned by**

**(Dr. Deepak Garg)**
Head
Computer Science and
Engineering Department
Thapar University
Patiala

**(Dr. S. S. Bhatia)**
Dean(Academic Affairs)
Thapar University
Patiala

## ACKNOWLEDGMENT

The successful completion of any task would be incomplete without acknowledging the people who made it possible and whose constant guidance and encouragement secured the success.

First of all I wish to acknowledge the benevolence of omnipotent God who gave me strength and courage to overcome all obstacles and showed me the silver lining in the dark clouds.

With the profound sense of gratitude and heartiest regard, I express my sincere feelings of indebtedness to my guide **Dr. Parteek Kumar**, Assistant Professor, Computer Science and Engineering Department, Thapar University for his positive attitude, excellent guidance, constant encouragement, keen interest, invaluable co-operation, generous attitude and above all his blessings. He has been a source of inspiration for me.

I am grateful to **Dr. Deepak Garg**, Head of Department, and **Dr. Ashutosh Mishra**, P.G. Coordinator, Computer Science and Engineering Department, Thapar University for the motivation and inspiration for the completion of this thesis.

Last but not the least I would like to express my heartfelt thanks to my parents and my friends who with their thought provoking views, veracity and whole hearted co-operation helped me in doing this thesis.



**(Umesh Kumar)**

**801333029**

# ABSTRACT

Now a days, the NUI is already popular in the form of multi-touch screens, which found its place in portable devices like Mobiles, Laptops, to Selecting objects, handling with multimedia and images with the help of multi touch interface.

The Touch less NUI converts the body Gesture to command for Computer. Nowadays, everybody is using mobile phone and computer as a very important gadget in their life. But there are some physically challenged people who are blind and the use of mobile phone or computer like device is very difficult for them. So,

There is an immense need of a system which works on body gesture or sign of sign language and speech as input, so there is the need to develop Touch less NUI which can recognize the body gesture and speech command, In order to solve the problem statement, objectives have been framed for this thesis work. And use the Microsoft Kinect Sensor and SDK V2 and HTK. Which can recognize the object, location of object, motion of object, shape of object human body. HTK provides the interface to implement the HMM model which has been used to implement Speech recognizer system. Developed Touch lees NUI provide interface to control the cursor and click operation just by waving hand gesture and Speech command.

# Table of Contents

# LIST OF FIGURES

# Chapter 1
# INTRODUCTION

Computers have progressed and extended into each and every arena of world. Be it home, office, school or college computers have become an essential part of our life. Nowadays, when anybody says about working the computer, anyone generally think, He is typing on the keyboard and touching the mouse device on the table to operate computer. All these types of input devices and methods have been invented and designed in 1960s as a kind of controlling devices and method allowing operators or user to control and use computers. In these days the technological advancement is making significant development in sensing method and technology to interact computer, which makes it possible to progressively replace the way of human-computer interface by adoptable natural interactions, which called as "Natural User Interface (NUI)".

Now a days, the NUI is already popular in the form of multi-touch screens, which found its place in portable devices like Mobiles, Laptops, to Selecting objects, handling with multimedia and images with the help of multi touch interface. Which makes the human-computer interface more easy/simpler like natural interface than traditional interface like mouse or keyboard. But, in the past decade the evolution and invention of the sensing technology has gone rapidly beyond the restrictions of the presently used human-computer interface. This technological advancement makes a great impact on computer vision/image related research. It helps the computers and devices to distinguish and detect and track the movements and gesture of the human body and object.

One of the landmark evolution took place on November 2010. The Microsoft launched MS-KINECT for Xbox 360, which tremendously changed the world of gaming in the form of the new touch–less interaction interface has released a flow of innovative solutions and ideas in the field of gaming and has also extended its usage to the shopping industry, entertainment industry, advertising industry, healthcare industry etc. It has revealed the new possibilities in front of the world that so far are only present in imaginations and sci-fi movies.

The goal of this thesis is to give a design and implement the touch–less Natural user interface with the help of the Microsoft MS-KINECT for Windows device and also gives possible numerous approaches in different designs of the touch–less interactions.

## 1.1 Types of NUI

The interface between computer device and human has always been the crucial objective of research and development ever since first computer device was developed. The very first computer, a complex interface was provided for interaction, which consisted of numerous big buttons and the response came in the form of combination of lights to the user,

The first human-computer interface was a Command Line Interface (CLI). Which allowed operators to gives the command to the computers more easily by just entering the commands with the help of a keyboard, which is not comfortable to untrained user, which led an evolution in the form of Graphical User Interface (GUI) facilitating untrained users to learn and use complex applications by graphical representations.

The GUI gave an idea to invent and design the mouse like devices which allowed the user to use through GUI and execute the most of known commands by just the click of mouse. Fortunately, today still we are using this tradition way of the human-computer interaction today, but in recent decade the evolution of the human-computer interaction is engaged to a more natural way for using computers like devices. Currently, most adoptable NUI is Multi-touch NUI which is mostly used in mobile, ATM, etc. Recently multi-touch NUI also introduce and which is a very new feature in Laptop and will change the whole scenario. But touch-less NUI is one of the crucial issue to solve. The description about Multi-touch and touch less NUI has been given.

### 1.1.1 Multi-touch NUI

The multi-touch interface or multi-touch NUI allows natural interface by touching the touch screen with the fingers (or special design stick). If its comparison is done with cursor-based interface, the user doesn't have to adjust and move the cursor to specific GUI presented location to select an item and click to open it. The user just touches specific location on touch screen which is graphically visualize by GUI. Which makes it more intuitive way to select item, rather than moving the mouse on the table. Moreover, because of the Multi-touch have the capability to recognize the two or more or multi touch points simultaneously on the touch screen, which gives an advanced functionality such as multi touch to zoom or evoking predefined actions [1].

The multi-touch NUI enables interaction with the help of number of touch point and their relative motion on touch screen which is predefined pattern. For example, user can tap on the touch screen in order to select or open an item or application, do a zoom, drag item use multi touch and relative motion on screen. This is one way of interaction to touch screen with the help of stick or natural finger by touching and moving. This is giving an improved natural feel to the ultimate interaction. Even though the multi-touch interface also refers to NUI, the interfaces for such technology are deliberate as a traditional GUI.

### 1.1.2 Touch–less NUI

The Touch less NUI means interaction between the human and computer devices with the help of Body gesture, motion, and voice command without touching of any in input devices. The improvement in capability of sensors in depth sensing within real-time, which clears the ground for touch-less NUI, has allowed the computer devices to visual capability especially without any the need of complex visual and image processing and analysis. The ability to recognize body movements creates a path for researcher to the design and implement of an entire innovative kind of human-computer interface, as The Touch-less Interface (Figure 1.1) [2].

Figure 1.0. Touch Less NUI working environment using MS-KINECT [3]

## 1.2 Challenges to achieve NUI

The interface between computer device and human has been all the time crucial objective of research and development since first computer device were developed.

There are the area which comes in consideration when anyone think about touch-less NUI like speech recognition, speech synthesis, image processing. These fields have its own challenges. To develop Touch-less NUI these all area should come together which is it's self a challenge.

### 1.2.1 Challenge in speech recognition and synthesis to develop NUI

It is one of the challenging areas to develop the universal touch-less NUI because of there so many native language are there. If anybody one want to build voice command interface they have to first develop specific speech recognition and synthesis system for targeted user because they have not common understandable language. Still not only NUI this is one most impotent aspect in natural language processing [4].

### 1.2.2 Challenge in Image processing to develop NUI

To develop touch less NUI, there should be system with visual capability. Before, the depth sensing devices came under consideration the visual capacity of computer system is only based on 2d color imaging processing which has its own limitation to give visual capability such that system can recognize on 2d gesture and object it is was the one of the biggest challenge in touch-less NUI. But now a day there are such technology that fulfill these gaps. One of them is MS-KINECT which have capability to capture not only color image, it can also capture depth image which allows to develop a system to which can recognize the object, location of object, motion of object, shape of object.

Now a day, the system are available which can recognize the object, location of object, motion of object, shape of object. There is a need to work on gesture recognition algorithm which have capability leaning with the help of Machine learning due the different NUI may have different way of gesture to control the system.

## 1.3 Tool used for development of NUI

As above it is clear that to develop touch-less NUI system. It needs the devices which can sense voice, image, and depth of user and can give the data to process. One of the most highly capable and suitable device for touch-less NUI is MS-KINECT which is a Microsoft product which also provide software development kit (SDK) to develop the gesture recognition and control application for window. Also give chance design and implement intelligent learning algorithm to develop gesture recognition. And for voice recognitions there is HTK which provide command line interface to implement HMM to develop speech recognition system

### 1.3.1 Microsoft-Kinect Sensor for window SDK

The **Microsoft-Kinect Sensor (MSKS)** has been launched and patented by Microsoft Corporation under a project "Natal" in 2006 [5]. The purpose behind it is to create an innovative game controller for Xbox-360. It was influenced by "Tokyo Game Show conference" in the 2005 in which the gamming console an innovative gaming-device was introduced with named as the Wii-Remote which was able to detect the three axes

movement and which encloses an optical-sensor that can easily identifies where Wii remote is pointing. Which was started the race of hacking of the device and Microsoft's Xbox division brought the in existence to start working on a competitive idea which can beat the Wii in future. Finally, the ultimate product has been launched and named as "MSKS for Xbox-360" and was constructed on basic idea of the Prime-Sense's depth sensing technology.

At starting, Microsoft develop only MS-KINECT device. It is MSKS for Xbox-360 gamming console and launched in public in November 2010. Once the MSKS gamming console was launched, User and researcher started hacking this device and as result a so many diverse scope and applications came under consideration and spread over the world of computer, Microsoft was on way to transform the existence technology to an entire new gamming and computer market. On the basis of technological needs, The Microsoft designed and launched an update version of the MSKS which is "MSKS for Windows", focused on the innovation and expansion for Personal Computer. Basically, there was only minor differences between both MS-KINECT versions with same capability; though, to commercialize PC Application for MS-Kinect Sensor, the official Software Development Kit (SDK) also had launched.

### 1.3.2 HTK tool Kit

The HiddenMarkovModelToolkit (HTK) is tool kit which is used to implement the Hidden Markov Model. HMM is statistical model based on hidden Markov process with hidden stats? HTK provides a command line interface to implement and manipulate HMM. It is a portable toolkit. HTK is mostly adopted in several research field like research of speech recognition, Speech synthesis, pattern recognition, based research, HTK provides of a set of library, modules and tools which are available in C source form. The tools provide most of command facilities for speech researches, like

(i). HMM training,
(ii). HMM testing
(iii). HMM results analysis, Gaussians and discrete distributions

# Chapter 2
# Review of Literature

## 2.1 Microsoft Kinect Sensor

The MS-KINECT has been launched and patented by Microsoft Corporation under a project "Natal" in 2006. It is a motion sense device. Which was develop for Xbox-360 game controller [5]. It was influenced by "Tokyo Game Show conference" in the 2005 in which the gamming console an innovative gaming-device was introduced with named as the Wii-Remote which was able to detect the three axes movement and which encloses an optical-sensor that can easily identifies where Wii remote is pointing. It started the race of hacking of the device and Microsoft's Xbox division brought the in existence to start working on a competitive idea which can beat the Wii in future. Finally, the ultimate product has been launched and named as "Microsoft Kinect Sensor (MS-KINECT) for Xbox-360" and was constructed on basic idea the Prime-Sense's depth sensing technology.

At starting, Microsoft develop only MS-KINECT device. It is MS-KINECT for Xbox-360 gamming console and launched in public in November 2010. Once the MS-KINECT gamming console was launched, User and researcher started hacking this device and as result so many diverse scope and applications came under consideration and spread over the world of computer, Microsoft was on way to transform the existence technology to an entire new gamming and computer market. On the basis of technological needs, The Microsoft designed and launched an update version of the MS-KINECT which is "MS-KINECT for Windows", focused on the innovation and expansion for Personal Computer. Basically, there was only minor differences between both MS-KINECT versions with same capability; though, to commercialize PC Application for MS-KINECT, the official Software Development Kit (SDK) also had launched.

### 2.1.1 Component of Microsoft Kinect Sensor

MS-KINECT is a sensing device with different type of sensor like Color image camera, Depth image sensor, and an array of microphone. The MS-KINECT is the first device which came with depth sensing capability along with color image camera. So MS-KINECT is mainly based on a color image technology with new depth sensing technology. To full file their sensing capability MS-KINECT includes following component that as shown in Figure2.10 are.

i). *Color image camera*, for capturing color image.
ii). *Infrared emitter* and *infrared depth reviser*, for capturing depth image.
*iii). An array of micro phone.*
*iv). LED for sensor running status indicator*

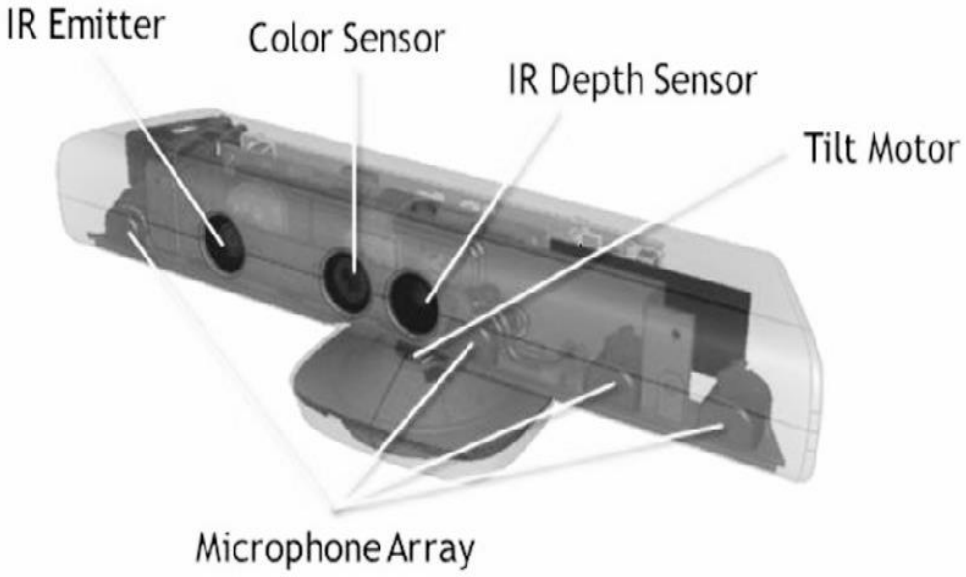



Figure 0.1 MS-KINECT components [6]

### i). Color Image Camera

Color Image camera is used for capturing and processing the color image stream data. Color camera can capture the color image data stream in different resolutions and formats. So color image stream data can store and capture in different data format MS-KINECT uses data formatting which is encoded by RGB. The RGB (red-green-blue) format representation uses 32-bit for the color image, and the unsigned linear format is X8R8G8B8– representation where 8 bites are fixed and revered for each color. Fps for color image frame in RGB format is 30 frame per second for 640*480 resolution and at 12 fps for high-definition 1280*960 resolution [7].

### ii). Infrared Emitter and Reflected Infrared receiver

IR emitter and an IR receiver sensor provides the capability to capture depth data. It happens only when they work together. The IR emitter is basically an IR projector. It is used to project or emit infrared light which is a pattern of "Random dot these dots are invisible to human eyes" toward direction of camera. But IR depth receiver can sense these dots. Reflected IR-receiver can sense the reflected dotted light from different objects, and translates them into the depth data which is a distance between the object and the sensor.

To capture depth data, infrared emitter get signal to start emitting the infrared light and at same one another signal goes to the reflected IR receiver to start capture the depth data within the range of the sensor the PrimeSense chip start work on captured data, and produces a frame stream as the depth image stream . The depth data means depth matrix of each pixel which is embodied of the data frame made up of pixels and there corresponding distance in millimeters which is calculated by taking camera plane as frame of reference

Combination of Pixel and distance gives the depth data segment. The data segmentation is a data which helps to relate the tracked skeleton, this association with the depth information used for body tracking. 16–bit unsigned integer format is used to represent the depth data where, the first 3 bits of data are fixed for the segmentation data and the remaining 13 bits for the depth information means for distance. As the distance

store in millimeter so that the maximal distance that can stored in the depth data can be up to 8192(2^13) millimeter which is approximately 8 meters. The range of depth data is illustrated by the (Figure 2.2.)

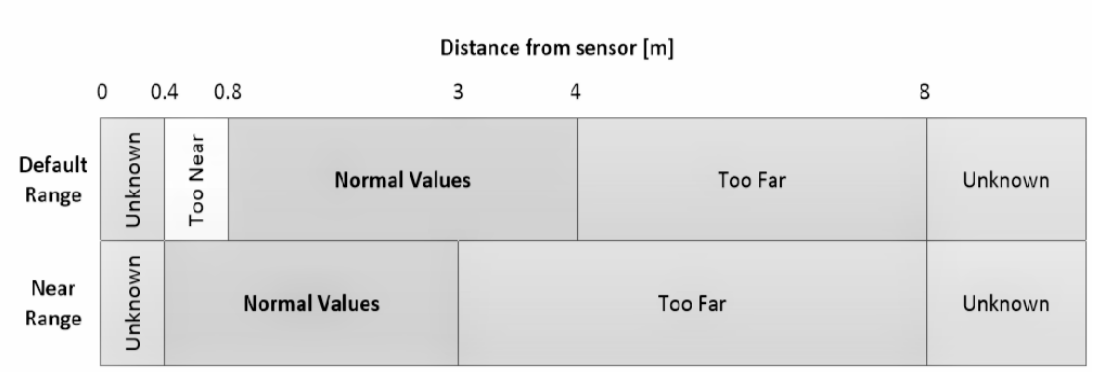



Figure 0.2 Depth sensing range of Kinect Sensor

Different resolutions for depth fame is possible which depends on number of pixel in frame. Fps for depth image frame is 30 frame per second for all resolution [8]. The available resolution for depth data frame are 640*480 pixels, 320*240 and 80*60 pixels.

The depth camera can carputer data frame at maximum distance 8 meters also it requires minimum distance is 0.3 meter to 0.4 meter. So depth camera can capture the frame in two mode that is the default and the near mode. The default made range is 0.8 meter to 4.0 meter. Which is the best range to capture the frame, near rage is 0.4 meter to 3.0 meter. According to capability of depth camera the depth space range can be .03 to 8.0 meter in both modes [18]. But, the quality of the depth frame come in the default mode which range is 0.3 to 4.0 meters. Value of 3.0 meters in near mode may be corrupted with distance.

### 2.1.2. Microsoft Kinect Sensor SDK

There was numerous Software Development Kits (SDK) developed and launched for allowing development of PC based custom application with help of the MS-Kinect Sensor. “Libfreenect library” is the first one. It was the result of the hacking the MS-Kinect Sensor device work in 2010, Up to this time Microsoft had not launched drivers to public use. The Libfreenect library enclose MS-Kinect Sensor drivers and allow a capturing of a depth and color data stream from the MS-Kinect Sensor device [9].

Another SDK, also launched before the Microsoft official one, is "OpenNI" which also released in 2010, just after a month of the releasing of MS-KINECT for Xbox-360. The OpenNI is a SDK library was released by the Company "PrimeSense", which is used the technology used by MS-Kinect Sensor Device. The SDK supports all standard as "Libfreenect library" support with extra capability such a Skeletal Tracking.

The Microsoft's official first release SDK version 1.0 for window was on February 2012 Currently, there is released latest version of the SDK, and is SDK version 2. And also launch MS-KINECT Xbox one for window which release on 2013. A step by progress and features of the SDK are explained in the following chapter [10]. The MS-Kinect Sensor SDK V2 provides facility develop application in different programing language with the help visual studio .net frame work.

### 2.1.2 Versions

The MS-Kinect Sensor is new technology for the programmer. There two version of MS-KINECT is launched. Both can use for Xbox gamming console and window PC.

i). First one is MS-Kinect Xbox-360 Sensor

ii). And second is MS-Kinect Xbox-one Sensor

For Xbox360 it doesn't need any extra power but for window pc it requires a external power supply and USB extension to connect

**i) MS-Kinect Xbox-360 Sensor**

The MS-KINECT has been launched and patented by Microsoft Corporation under a project "Natal" in 2006. After launching of MS-KINECT the several version for MS-KINECT-SDK for window has launched. The first version of SDK is Beta-1 version which was launched in Jun 2011. Which found a great response from developer community. After this another updated version was launched in November-2011. Which was MS-KINECT-SDK Beta-2 for window. These two version was not commercial. After success of these version, Microsoft came with its first commercial version in Fab 2012 that was MS-KINECT-SDKv1.0. also update the MS-KINECT hardware. As show on so many feature was launched. Last version of this series was launched in sept-2013 with MS-KINECT-SDKv1.8 [11].

### ii) MS-Kinect Sensor Xbox-One

This is the latest hardware version which was launched in 22 nov-2013. It is successor of Xbox-360. This version of sensor update theirs's hardware capability range of sensing etc. Also launched the SDKv2.0 for window which is the latest available version. There are the top changes have introduce from SDK version1.8:

- Sensor provides much better image resolution for both color image and IR image
- Number of joints increases
- Gives library for Face gesture and expressions
- Can detect hand states like close, Open etc.
- And also improved accuracy sensor.
- Windows_Store_app_support

The MS-KINECT SDK also include the driver for hardware to control the component of MS-KINECT like color camera, depth image camera, array of microphone. Basically MS-KINECT handle the three data streams comes from different data source these are the following types:

- Color data stream
- Depth data stream
- Audio data stream

### 2.1.3 Main Feature

The MS-KINECT SDK provides an inbuilt library which helps to directly access to the sensor components, such as Color Image camera, Depth image camera (inbred data) and the array of microphone etc. Programmer can even extend at own level. There is a list of operations which can be perform with the help of MS-KINECT SDK. Which provides a background to implement [12].

i). Access to capture and process to the data stream for color image

ii). Access capture and process the data stream for depth image

iii). Access to the infrared stream

iv). Detect human body and four leg animal body and Track.

v). Creating human skeleton and detect body joint

vi). Body gesture recognition

vii). Assess to the audio stream etc.

## 2.2 HTK Tool Kit

The HMM.Toolkit (HTK) is tool kit which is used to implement the HMM. HMM is statistical model based on hidden Markov process with hidden states [13]? HTK is provides a command line interface to implement and manipulate HMM. Which is a portable toolkit. HTK is mostly adopted in several research field like research of speech recognition, Speech synthesis, pattern recognition, based research, HTK provides of a set of library, modules and tools which are available in C source form. The tools provide most of command facilities for speech researches, like

(iv). HMM training,
(v). HMM testing
(vi). HMM results analysis.
(vii). Gaussians and discrete distributions

Which helps the researcher to developed own complex application specific HMM systems

# Chapter 3
# Problem statement

The Touch less NUI is converts the body Gesture to command for Computer. Nowadays, everybody is using mobile phone and computer as a very important gadget in their life. But there are some physically challenged people who are blind and the use of mobile phone or computer like device is very difficult for them. So, there is an immense need of a system which works on body gesture or sign of sign language and speech as input, so there is the need to develop Touch less NUI which can recognize the body gesture and speech command, In order to solve the problem statement, objectives have been framed for this thesis work. These have been discussed in next section.

## 3.1 Objectives

The main objective of this thesis is to understand, implement and explore the NUI technology. NUI related research needs the sophisticate and special devices which have the capability to detect the Human body and their gesture. To develop a Basic NUI following Objectives are proposed and used.

1. To study the existing system of NUI technology and their related state of art.
2. To study the Microsoft Kinect development frame work and its implementation.
3. To develop Hand gesture controlled cursor NUI to select and click on items with the help of Microsoft Kinect sensor and SDK v2.
4. To study the HTK (HMM tool kit) based architecture and acoustic model.
5. To implement the HMM model and architecture to detect Speech based command for NUI.
6. Test the developed Touch Less NUI and propose there future scope.

## 3.2 Methodology

To fulfill all the objectives, which are discussed in previous section, the following methodologies have been used.

i). The Literature survey has been done by study of existing state of art of NUI Technology like Multi-touch NUI and Touch-less NUI and their challenges and requirements.

ii). The Microsoft Kinect senor and its SDK has been installed which provides various essential data and there access to develop application which can detect human gestures

iii). With the help of body frame and coordinate mapping facility of Kinect sensor and SDK, The window WPF application has been developed which cans control cursor and perform Click operation.

iv). The literature survey has been done to understand the state of art of Speech recognition system.

v). Cygwin, HTK, Julius, audacity tools have been installed to implement Speech command system.

vi). A Touch less NUI has been develop and test, through which, computer cursor can be control and can perform the click operation

# Chapter 4
# Implementation

This chapter describes an implementation of the NUI .The implementation of the NUI consists of data layer for sensor's data processing and representation, Touch-less interactions using hands for moving cursor and a several kinds of ways for performing action, Implementation of NUI describe in two segment of this chapter, which are

(i). First part of Implementation describes the design and Implementation NUI using MS-KINECT.

(ii). Second Part of this chapter describes the implementation of speech command recognition using HTK.

## 4.1 The design and Implementation NUI using MS-Kinect

The implementation the touch-less NUI, requires to develop a H-G (Hand Gesture) control application using MS-Kinect. Whole development process of MS-Kinect bases application described following step by steps

### 4.1.1 System Requirements for implementation

To starts development of MS-Kinect based application or NUI, needs a specific environment setup which includes MS-Kinect for window, MS-Kinect SDK with supported operating system version, a specific PC-CPU configuration. All these requirement are describe in two category as follows [14].

### i). Hardware requirements:

The MS-Kinect Xbox One Provides the High quality data stream and real time interaction to real time data so it is required to Window PC should have the minimum specific hardware capability which is shown in Table 3.1.

Table 0.1 Hardware specification to support MS-KINECT

| |
|---|
| • 64-bit (x64) processor |
| • 4 GB Memory (or more) |
| • Physical dual-core 3.1 GHz (2 logical cores per physical) or faster processor |
| • USB 3.0 controller dedicated to the Kinect for Windows v2 sensor* |
| • DX11 capable graphics adapter** |
| • A Microsoft Kinect v2 sensor, which includes a power hub and USB cabling |

### ii). Software requirements

The MS-Kinect SDK v2.0 provides an environment for developers to develop applications. But to install SDK. Following requirement should fulfill, details are given in Table 3.1.

Table 0.2 Software requirnmnt

| | |
|---|---|
| Version | 2.0.1410.19000 |
| File Name: | KinectSDK-v2.0_1409-Setup.exe |
| Date Published: 21-10-2014 | File Size: 275.8 MB |
| Supported Operating System | Windows 8, Windows 8.1 show on |
| SDK 2.0 includes the following: | • Drivers for using Kinect v2 sensors on a computer running Windows 8 (x64), Windows 8.1 (x64), and Windows Embedded Standard 8 (x64)<br>• Application programming interfaces (APIs) and device interfaces<br>• Code samples |
| Software Requirements | • Visual Studio 2012 or Visual Studio 2013 |

re

### 4.1.2 Installation of the MS-KINECT SDK

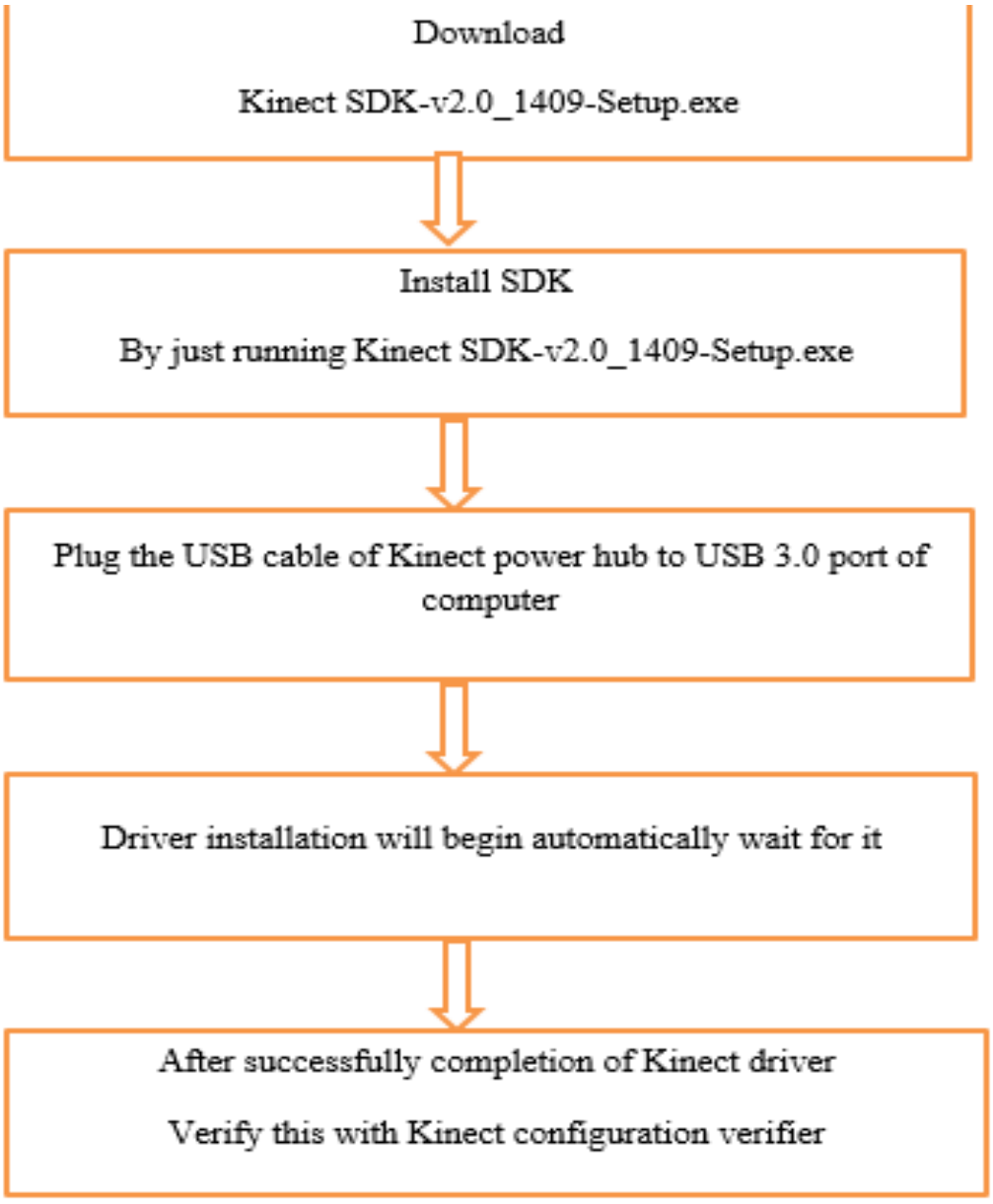



Figure 0.1 Steps to install Kinect SDK and driver

### 4.1.3 Architecture of MS-Kinect Sensor / SDK

The MS-Kinect-SDK gives a refined software development library and tools to develop application MS-KINECT-based with help of usual input, which have the capability to sense and responds to physical natural action. The MS-KINECT and the SDK library interface with an application, given in Figure 4.2 [15].

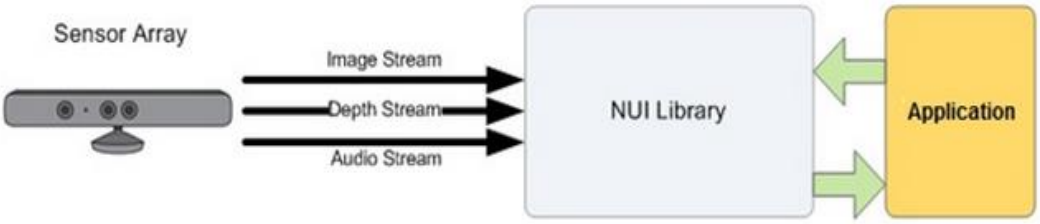



Figure 0.2 MS-KINECT and MSk-SDK interface with an application [15]

These components include the following:

The hardware parts of MS-Kinect, with are

i). Kinect sensor.
ii). USB 3.0 extension for Kinect Sensor to connect to with the computer.
iii). Power adapter to Kinect Sensor.

i). **MS-Kinect Sensor** drivers – MS-Kinect Sensor driver is the part of SDK.As SDK is installed. There is needed to plug in the sensor which automatically starts installation of the driver installation. Due to MS-Kinect device is a combination of different sensor so its drivers support, Array of microphone array, Color image Camera, Infrared camera which can access with the help of SDK provided different APIs in Windows 2 of Figure 4.3.

ii). NUI API for Components MS-Kinect Sensor to interact with skeleton tracking data, voice data, and color image data and depth imaging data 3 of Figure 4.3.

iii). There is a DirectX-Media Object (DMO) for array of microphone to voice beam formation and to detect location of Voice source 4$^{th}$ of Figure 4.3.

iv). Windows standard native API for development 5$^{th}$ of Figure 4.3.

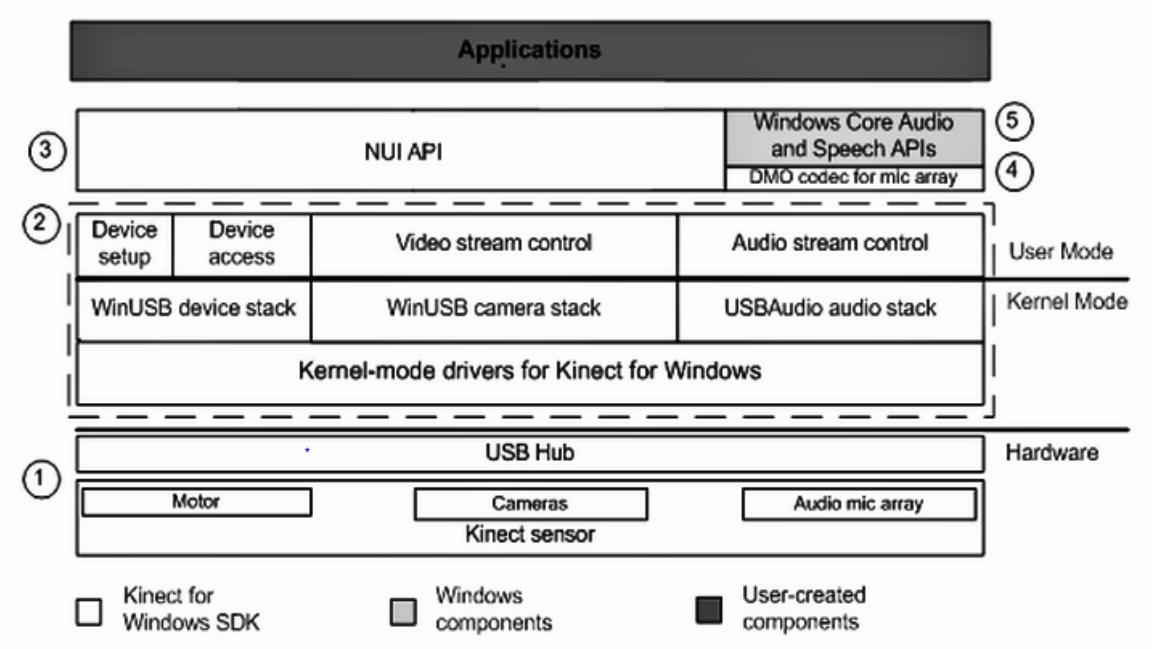



Figure 0.3 MS-KINECT detail Architecture [15]

### 4.1.4 Data source

MS-Kinect Xbox one (SDK v2) provides five types of input data streams with help of color camera, IR projector-receiver, array of micro phone and SDK [16]. These are the following data streams,

i). Color image data stream.

ii). Depth image data stream.

iii). Infrared image data stream.

iv). Body index data stream.

v). Audio stream.

These data streams are the Data source to develop MS-Kinect based application. And MS-KINECT SDK provides MS-KINECT ’drivers to control sensor, MS-Kinect Sensor run time environment to run MS-Kinect abased application, physical access to data stream, also provides .net framework,

### 4.1.5 Steps to Develop MS-KINECT WPF application

To create develop window MS-KINECT based application using visual studio 13 describe as following

**Step1. Set visual basic 2013**

i). Start with visual studio, create “New application”.

ii). Set the different properties

a) Capabilities: enable the capability like microphone, webcam. These are mandatory capability to enable for MS-KINECT app.

b) Add new references: there is need to add dll files as references. For MS-KINECT apps there is mandatory to add the “windowsPreview.MS-KINECT”.

c) Check configuration properties to set correct platform according CPU X64 or X86

iii). After all use the primary namespace like “windowsPreview.MS-KINECT” as requirement. There should be MS-KINECT lib, which provide MS-KINECT based functionality.

**Step2. Initialize the MS-KINECT sensor:-**

The MS-KINECT based application are the real application and based on MS-KINECT data, which is capture by the MS-KINECT so there should initialize the sensor at the starting of program

MS-KINECT SDK provides the “kinectSensor” class which is the only interface to initialize the MS-KINECT and always use. As shown in Figure 4.4

```
C#
this.sensor =  KinectSensor.GetDefault();
this.sensor.Open();
// Make the world a better place with Kinect
this.sensor.Close();
```

Figure 0.4 To KinectSensor class

**Step3. Select MS-KINECT Source**

i). The multiple source are provided by the KinectSensor class

ii). So KinectSensor class also provide metadata to access the these data sources as shown in Figure 4.5

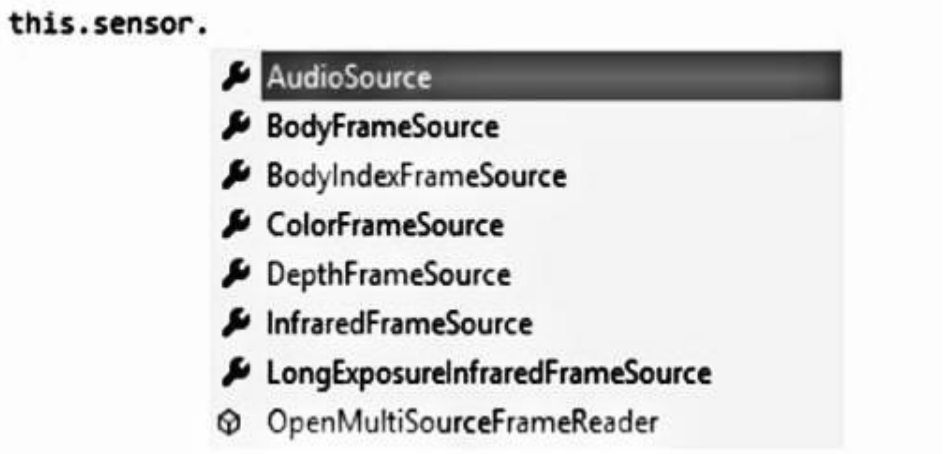



Figure 0.5 MS-KINECT data source

**Step4: Create event as start capturing the data stream**

i). Data streams are provided by the data source of the MS-KINECT, Therefore it requires the Reader to get access the these frames

ii). With the help of the Reader handles the arrived frame as creating even of capturing the event As show in figure

iii). Given in the Figure 4.6, there is infra-red frame reader "reader", which creates infrared frame arrived event as frame is came.

```
InfraredFrameReader reader = sensor.InfraredFrameSource.OpenReader();
reader.FrameArrived += InfraredReaderFrameArrived;
...
```

Figure 0.6 Frame Arrive Event

**Stepe4: access to frame with the help of Frame Reference**

i). Data stream is sequence of the frame, so they come in sequence which associate with the particular time interval.

ii). Which is send with the event argument to handle the particular frameArrived event as shown in Figure 4.7

iii). Which gives the access to actual frames as shown in the Figure 4.8

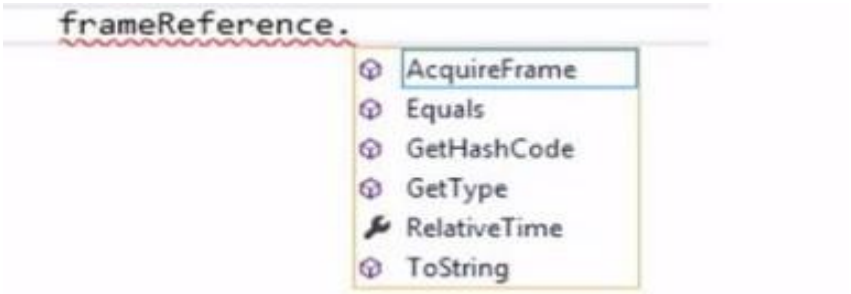



Figure 0.7 To get data Frame stream

iv). Here doesn't direct access to frame because of the 15-30 frame is came per second so particular frame belong to particular event ,these frame reference gives the clearance to frame coming event.

```
void irReader_FrameArrived(InfraredFrameReader sender,
                           InfraredFrameArrivedEventArgs args){
    using (InfraredFrame frame = args.FrameReference.AcquireFrame())
    {
        if (frame != null)
        {
            // Get what you need from the frame
        }
    }
}
```

Figure 0.8 Frame arrived event handler

**Step5: Now get the actual data "Frame"**

i). It is the actual data to handle and manipulate like copy the frame

ii). There is metadata (properties) to each frame like color: format, width, height, etc.

iii). This give to real time manipulation of data.

These the five step to develop MS-KINECT application

#### 4.1.5.1 Capturing the Color streams

To capture color stream the same five steps are follow which was described in 4.1.5 section. With the help of ColorframeSource Figure 4.5 metadata of MS-KINECT's Sensor class.

The ColorFrame class is used to represent and implemented the color image data. The class give the access to the color image data with help of data format, image's resolution, capture time and frame reference etc. A byte array is used store and represent the image data. The color image can store many format like RGB,YUV or Bayer, The RGB (red-green-blue) format representation uses 32-bit for the color image, and the unsigned linear format is X8R8G8B8– 32 bit representation where 8 bites are fixed and revered for each color. Fps for color image frame in RGB format is 30 frame per second 640*480 resolution

(30*640*480*32 bit per second or 37MB per second data stream) and at 12 fps for high-definition 1280*960 resolution. The ColorFrame class also provides method or function to manipulate pixel data to implementation several algorithm like tracking algorithm to find out 2D coordinates with help of color image with respect to sensor

#### 4.1.5.2 Capturing depth streams

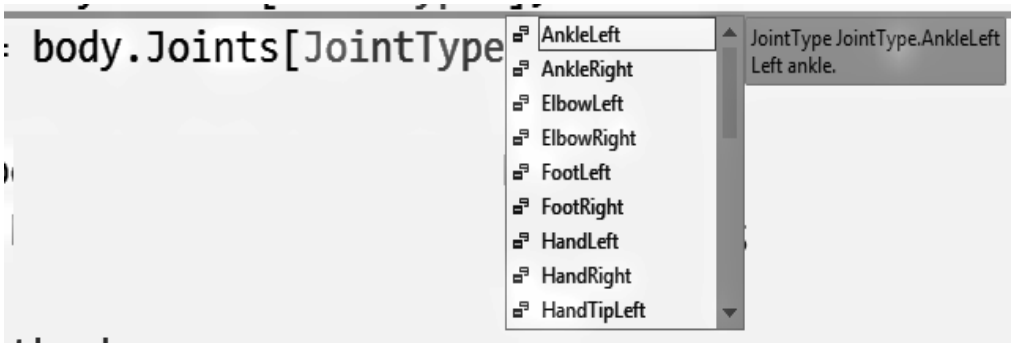
DepthFrameSource Class Figure 4.5 is used to design and implement with the depth image. The class give the access to the depth data with help of data format, image's resolution, capture time and frame reference, distance of each pixels etc. 16–bit unsigned integer format is used to represent the depth data where, the first 3 bits of data are fixed for the segmentation data and the remaining 13 bits for the depth information means for distance. If depth value -1 which also called as invalid depth value. Which means the pixel which have invalid depth value is not within the visual range sensor something is going wrong. The DepthFrame class also provides method or function to manipulate depth data to implementation several algorithm like tracking algorithm and finding 3D coordinates with help of color image with respect to sensor.

There all five step describes in section 4.1.5 are used to capture depth data with the help of DepthFrameSource class.

#### 4.1.5.3 Body Frame

This the most essential data source of MS-KINECT: Which allow to implement touch-less NUI. At starting MS-KINECT SDK provided the access to 20 body joints to track. In latest version of MS-KINECT SDKv2 provides this facility up to 25 joints. There are two new joints in each hand has included. One fists and thumbs. With the help of these two new joint it easy detect state of the hands like it is close or open. Even, due to the enhancement of capability of depth sensing, the tracking capacity also has been increases

To get body frame same steps are follows, which is described in the section 4.1.5. With bodyFrame metadata of kinectSensor class as shown in Figure 4.9.




MS-KINECT have capability to track up 6 body at time and for joint tracking up to two body. Figure 4.10 shows the all body joints.

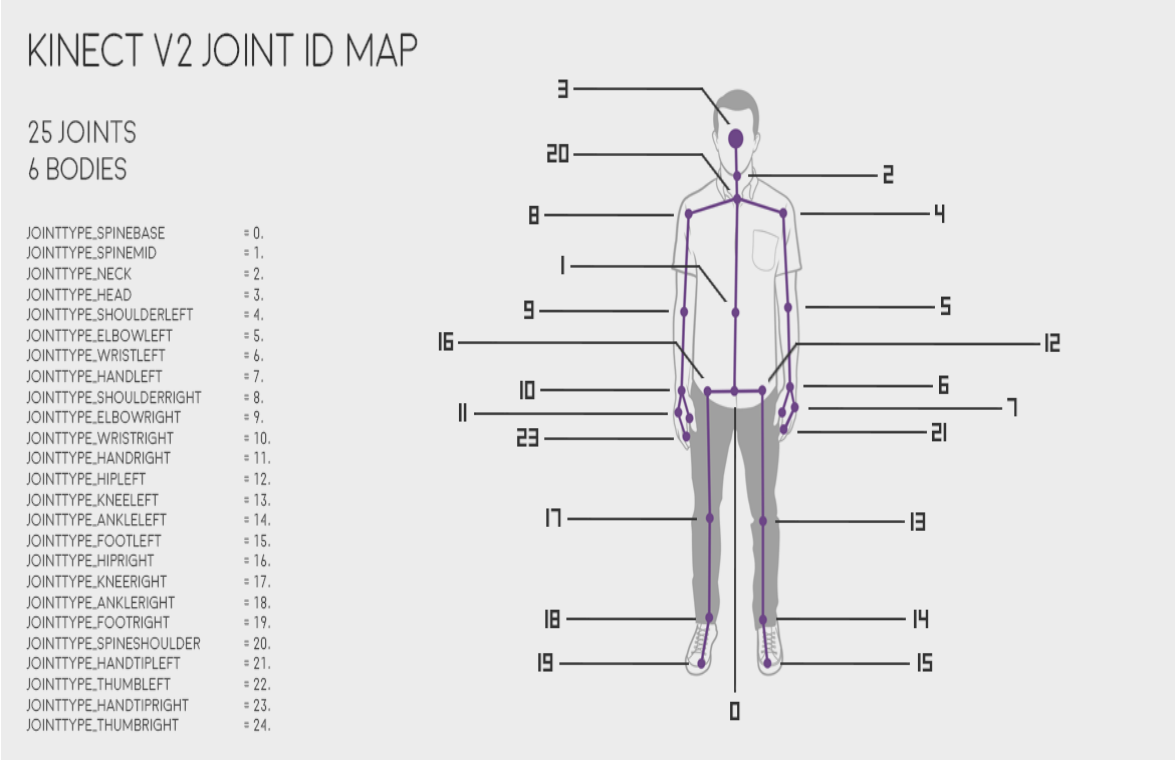



Figure 0.9 List of all body joint MS-KINECT [17]

### 4.1.5.4 Coordinate Mapper

MS-KINECT-SDK also provides the CoordinateMapper utility or property of MS-KINECT Class. It provides way to detect a point in three-D space which is corresponding to two-D image. So it can use with very MS-KINECT instances.

There is Example shown in Figure 4.11. It is write to get (X, Y, Z) mapping of each body Joints. These value are in meter.

```
foreach (Joint joint in body.Joints)
{
    // 3D coordinates in meters
    CameraSpacePoint cameraPoint = joint.Position;
    float x = cameraPoint.X;
    float y = cameraPoint.Y;
    float z = cameraPoint.Z;
}
```

*Figure* 0.10 3D Coordinate for body Joint

### 4.1.5.5 Calculating the distance between two joints

Body-data is provided in three-D by MS-KINECT. However, with the help of coordinate mapper utility of MS-KINECT, it is possible to get 3d coordinate (X, Y, Z). Which was discussed in section 4.1.5.4.

By using Pythagorean Theorem, it is very easy to calculate distance (D) between two bodies joint after getting the coordinate of two joint. Example there is two joint coordinate are (X1, Y1, Y2) and (X2, Y2, Z2) due to the these coordinate value are in meter so calculate distance value will be in meters as shown in (4.1).

$$D=\sqrt{(X1-X2)^2+(Y1-Y2)^2+(Z1-Z2)^2} \quad \ldots\ldots\ldots\ldots\ldots (4.1)$$

### 4.1.6 Implement the gesture control touch-less NUI.

As previous, body frame section 4.1.6.4 and coordinate mapping utility of MS-KINECT class is discussed. So with the help of these it is possible to track anybody joint. One of touch-less NUI can archives

#### 4.1.6.1 Gesture's designing.

The natural user interface idea creates new way of interaction between human and machine. That enable the recognizer to recognize the gesture and pattern of gesture. Pattern of gesture means a way change in gesture. So particular gesture pattern allow the system to executing predefined action, but the efficiency of recognizer is also depends on human's experience and intuition. So gesture's design for NUI is a critical and important aspect. If the H-G is not correct, natural and reliable, the application should unresponsive and which may difficult to use the application to the users. So, there are several factors, issues and circumstances which should be tackled carefully for a natural, efficient, reliable gesture design to avoid users' inconvenience [18].

**i). Designing a Gesture**

A reliable and natural gesture design depends on scope of application and users its diversity depends on the user's analysis and institution to understand of a gesture that could be totally dissimilar from the different users. Criteria to design Gesture is

I. Gesture should be Natural and reliable.
II. Gesture should be flexible mean system should learn from the user's gesture.
III. If gesture is recognized means match to predefined gesture then perform action if not recognize then should not perform any action.

There is one more impact on the gesture's operation. Which has an option to use one or two H-G (hand gesture). One H-G is much more spontaneous and relaxed to do than two-handed, whenever a double handed gesture planned, both H-G should be identical. Single-handed gestures is used for critical and regular action

The gesture's design should be such way that, user should not tire or in exhaustion by acting on typical gesture constantly. If it is happened, it will have an undesirable experience and probably he will leave. There should be consideration of possible method to avoid such condition, so that, the fatigue is reduce, for this, besides of one H-G, there should be two H-G is design so if user is going to tired, He or she can switch hands. For successful NUI interaction these are the thing should consider, so it is required to take feedback from the users [19].

### ii). Wave gesture

Wave gesture is one of the most natural and common gestures. Normally human use waving the hand to calling someone by visually. In the NUI interaction, the wave-gesture can be design to use for indication that the user is in ready state to begin the interaction. Microsoft also use the hand wave-gesture and gives as an encouraging method of defining that user is intended for engagement with the system [18].

The hand-wave is a wave-gesture with natural actions which is easy to detect with the help of an algorithmic approach [20]. By observation of human activity like the hands waving it can be notice that we can make the association between the hand (palm) and the arm during the waving gesture. Where we can notice that waving gesture starts in neutral straight position when the A part of arm (wrist to E-J let's named as WE) is perpendicular (90 degree angle) to the remaining part of the arm (E-J to shoulder let's named as ES) by make E-J as center. If the WE's position changes a certain threshold by moving WE part of hand either to the right or to the left, it will be consider as a segment of the waving hand-gesture. To construct complete hand wave gesture, the WE part of arm should oscillates with respect to straight position of WE multiple times. Else it will not be consider wave-gesture. This gesture is easy to perform by anybody, and easily track with help of two joints of tracked skeleton as shown in Figure 4.12,

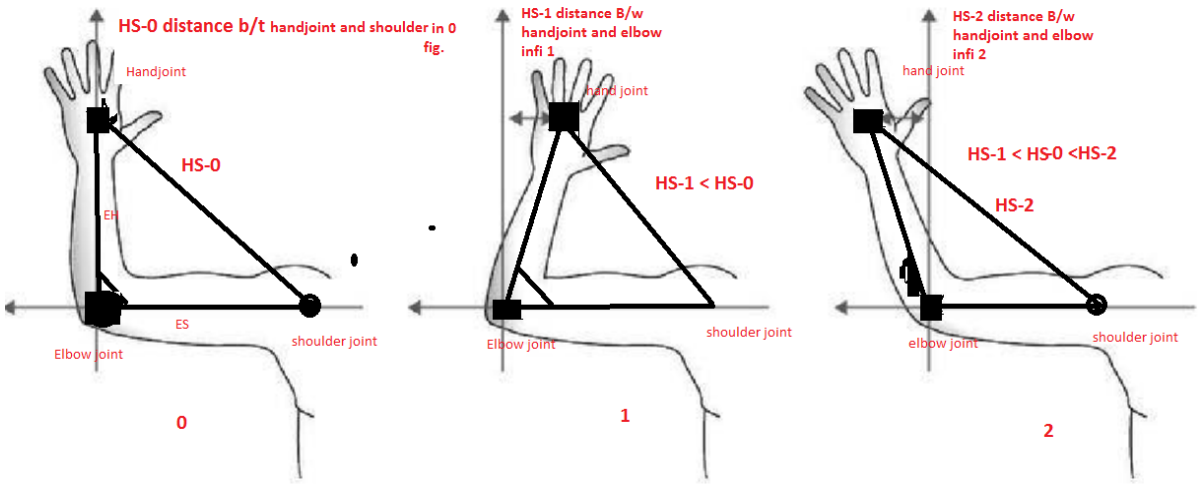



Figure 0.11 Hand Direction Calculation [10]

Algorithm 1 Hand-Gesture Recognition

**4.1 Algorithm** to Wave gesture Recognition

4.1.1 Bf = getframe(bodyFrame)

4.1.2 Get joint with help of bodyFrame Class

- Hj = getjoint( Bf.Righthandjoint )
- Sj = getjoint( Bj.Rightshouldejoint )
- Ej = getjoint( Bj.Rightelbowjoint )

4.1.3 Get the coordinate of joint with help of coordinate mapper

- CHj = coordinateMap( Hj )
- CSj = coordinateMap( Sj )
- CEj = coordinateMap( Ej )

4.1.4 Calculate Distance between each joint upcoming each frame

- HS = distance( CHj, CSj )
- HE = distance( CHj, CEj )
- ES = distance( CEJ, CSj )

4.1.5 Calculate threshold distance of HS

- HS-0 = $\sqrt{HE^2 + ES^2}$

i.1.6 If (HS == HS-o)

- Cursor(fix)

Else if(HS < Hs-0)

- **Cursor(Left)**

**Else**

- **Cursor(right)**

#### 4.1.6.2 Description of Algorithm:

The algorithm 4.1 starts from capturing the Body frames as (4.1.1). With the help of body frame to create H-G, it is needed to get three joints and these are H-J, E-J Joint and S-j as (4.1.2). Coordinate Mapper provides the coordinate of these joints as (4.1.3). For each Frame the distance between each joint in calculate as (4.1.4) and it can be notice that distance between E-J-shoulder (ES) and E-J-hand (HE) is always same. Only the distance between H-J and shoulder (HS) will chanced as shown in Figure 4.12. For right hand if the distance HS-0 is one threshold value which also fixed due ES and HS is fix.

As new frame will and these distance is calculate again, Due there is only a chance to change in HS if HS is less than HS-0, it will indicate a command to move cursor left and if greater than, it for right.

## 4.2 The implementation of speech command recognition using HTK

This section will discuss the implementation Speech command recognizer using HTK. As following

### 4.2.1 Requirements for implementation

There are some pre requirements for developing the Speech command recognizer. These have been given below.

**i) Hardware requirements**

To develop a Speech command system, it requires the essential devices, which are as follows.

- One of the most important device is good quality microphone. Which has been used for capturing and recording (saving) the speech. Due to, the recorded speech will use in training the system, recorded speech should be noise free. A sophisticated microphone is needed to serve the recording [21].

### ii) Software requirements

The software tools which have been used for developing the Speech-to-Text system are as follows:

- **Cygwin:** Cygwin tool provides the Linux based environment for window develop STT system.
- **Hidden Markov Model Tool Kit (HTK)** The HiddenMarkovModelToolkit (HTK) is tool kit which is used to implement the Hidden Markov Model.
- **Julius:** Julius is a speech recognition engine. Which executes the STT.
- **Audacity** This tool is used to record the speech files

### 4.2.2 Architecture of Speech-to-Text system

A hidden Markova Model based "Speech to text system" is implemented in four phase architecture; First phase for Data Preparation , second and third phase for training using Monophones and triphones, and fourth phase to execution the system. Abstract architecture is given in Fig. . Description of this architecture is explained phase wise in subsequent sections

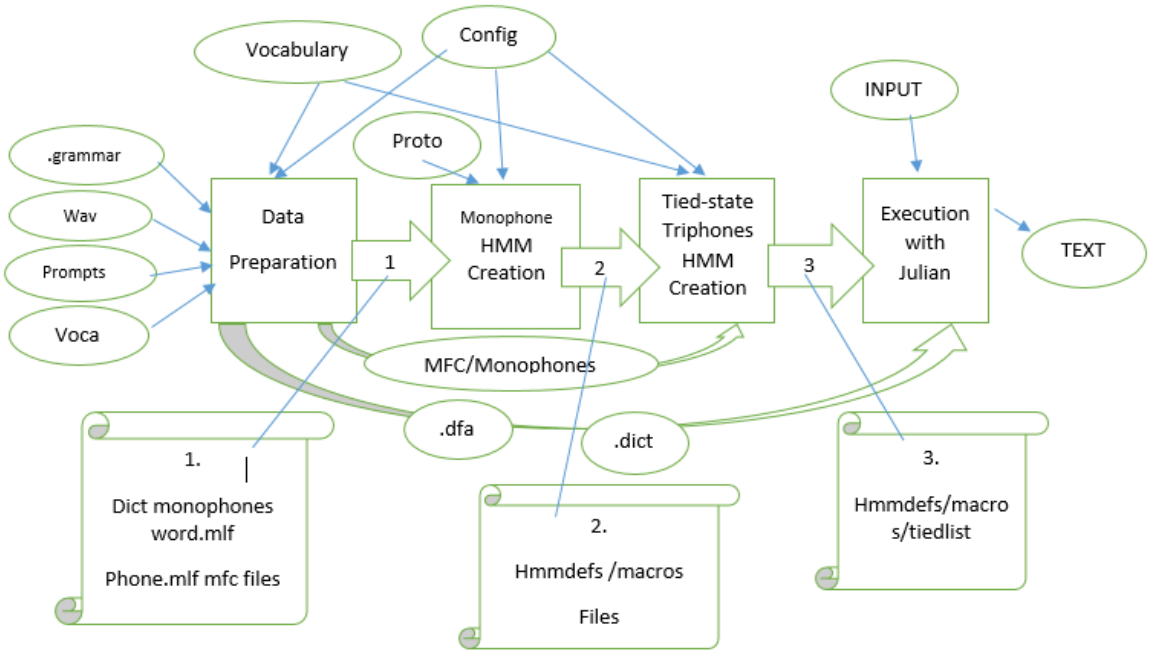



Figure 0.12 STT Architecture

### 4.2.3 Architecture

These four phases of the architecture of Speech command system are as follows:

- Data preparation phase
- MonophonesHMM Creation
- tied-state-triphonesHMM Creation
- Execution with Julius Interface.

The description of each of these phases has been discussed in subsequent sections

#### 4.2.3.1 Data preparation phase

In speech recognition, system needs data to train its acoustic model such as Grammar file, speech file, corresponding training text file and Vocabulary file. Some intermediate files are also created while processing of these files which are also required to train the system. The explanation of the files required for training is given as follows.

**i) Grammar File**

For recognition, grammar defined rule for what STT system is going to listen or recognize. ".grammar" file is used to define the form of input; isolated word or group of words. The ".voca" file contains a list of actual words. Grammar for recognizer is provided into two files ".grammar" and ".voca" file. The ".The description of ".grammar" file is given in Table 4.3

Table 0.3 Grammar and Voca file

| | |
|---|---|
| S : NS_B SENT NS_EB<br>SENT: DIGIT NAME …(1.a) | % NS_B <s> sil<br>% NS_E </s> sil<br>% DIGIT<br>ONE w ah n<br>TWO t uw<br>%NAME<br>ADVICE ae d v ay s<br>BOY b oy …(1. b) |

Basically .grammar file is based on Modified BNF (Backus Normal Form or Backus Naur Form) and the signature is SYMBOLE: [expression] Where SYMBOLE is always a Nonterminal and expression is sequence of terminals and nonterminal. As BNF, terminals represent a constant value, In Julian Grammar, these symbols represent the word category (list of words) defined in voca file. It is placed on the right hand side of the colon. As given in (1.a), S is starting symbol, NS_B, NS_E DIGIT and NAME are the terminals which represent word category. As given in (1.b), SENT is non-terminal, which can be replaced by right side expression where NS_B and NS_E represent the Silence that occurs at starting and ending of the words or sentences which STT wants to recognize. So, S, NS_B and NS_E are always present in a grammar file. Each word category in .voca file starts with character "%" followed by category name signature % [Word category] [Word] [Word pronunciation], NS_B, NS_E, DIGIT and NAME are word category and defined in .voca file.

**ii) Training text file:**

Also named as prompts file. On the basis of this file, audio files are created. Prompts file contains the list of the words which have to be recorded. This file is created in such a manner that the created pronunciation dictionary should phonetically balance. Some content of prompt file is given in Figure 4.14. Prompts file should contain at least 30-40 sentences "of 8-10 words. .

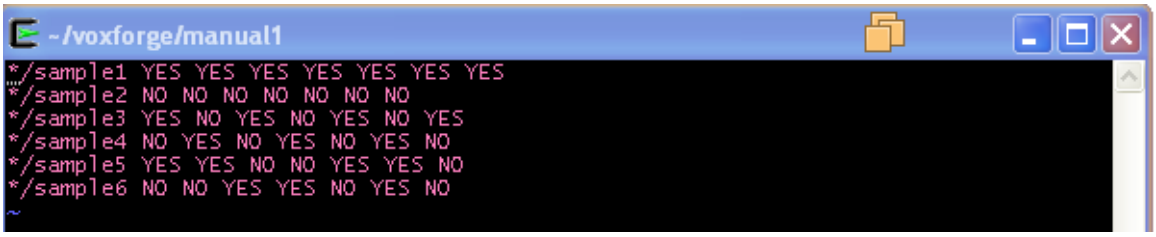



Figure 0.13 prompts File

**iii) Speech Files:**

Speech files are recorded in .wav format. These files are recorded with the help of 'Audacity' recording tool. Every sentence in prompts file corresponds to a speech file.

**iv) Vocabulary file:**

It is also called global dictionary. This file contains collection of all words with their pronunciation information. This file is used to create transcription training files. A view of this file is given in Figure 4.15

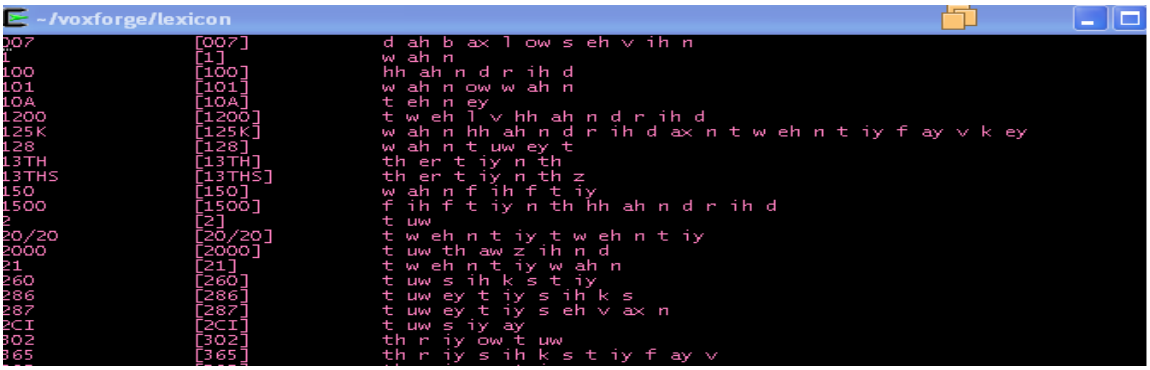



Figure 0.14 Dictionary File

### v) Creating pronunciation dictionary

To create pronunciation dictionary, it needs word list which is used in prompts file. The Perl script "prompt2wlist" is used to create wlist. After having wlist HDMan command is used to create pronunciation dictionary with the help of global.dedscript. It creates two files, dict file and monophone0 file. dict file have each word of wlist with corresponding pronunciation details and Monophone0 file contains list of all distinct phones as given in Figure 4.16.

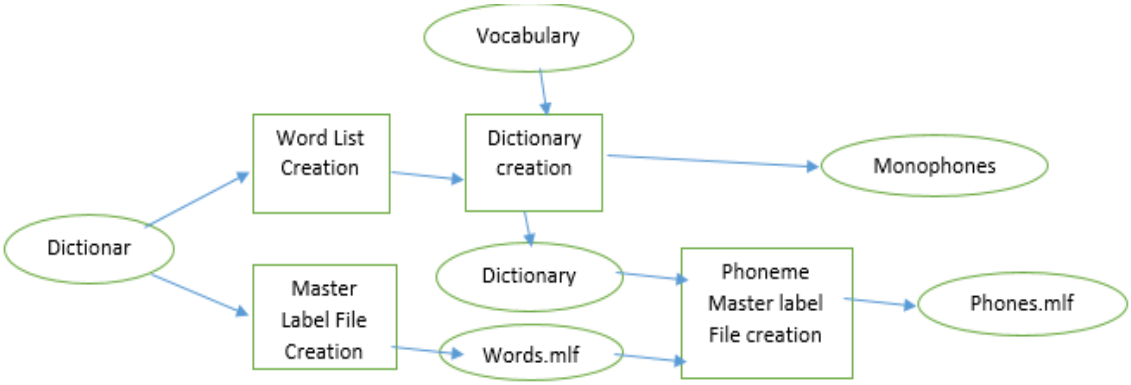



Figure 0.15 Pronunciation file creation

### vi) Creation of transcription files

HTK requires word label and phone label transcription files, which is used to train STT. These files are called as Master Label Files (.mlf). Basically, these files are created from Prompts file. Word label transcription (word.mlf) file is created by running" prompts2mlf" script, which rearranges the content of prompt file in word.mlf file, which contains each word of prompt file but in new line. Next HLEd command is to process the word.mlf with help of mkphones0.led script file and resulting phone0.mlf file is given in Figure 4.16.

### vii) Creation of mfcc files:

HCopy command is used to create the MFCC (.mfc) from .wav file with the help of configuration file, .config [8]. These MFCC (.mfc) files contain the feature vectors of recorded .wav files.

#### 4.2.3.2 Monophone HMM Creation:

This the first phase for HMM training that is used to create well-trained set of Single-Gaussian monophones. HMM training is based on Prototype Model which is defined in proto file. Proto file, .mfc files, monophone files[40], configuration file and phones.mlf files are required for creating HMM. There are number of monophones in HMM file which are distinguished based on five states, where state 1 and state 5 are opening and closing states, states 2, 3 and 4 have values for means and variance. At The starting point, HMM file will be a set of identical monophone HMMs in which mean and variance are identical. These are then retrained in sub-phases, namely, creating flat start monophones and re-estimation, fixing the silence models and realigning the training data as given in Figure 4.17.

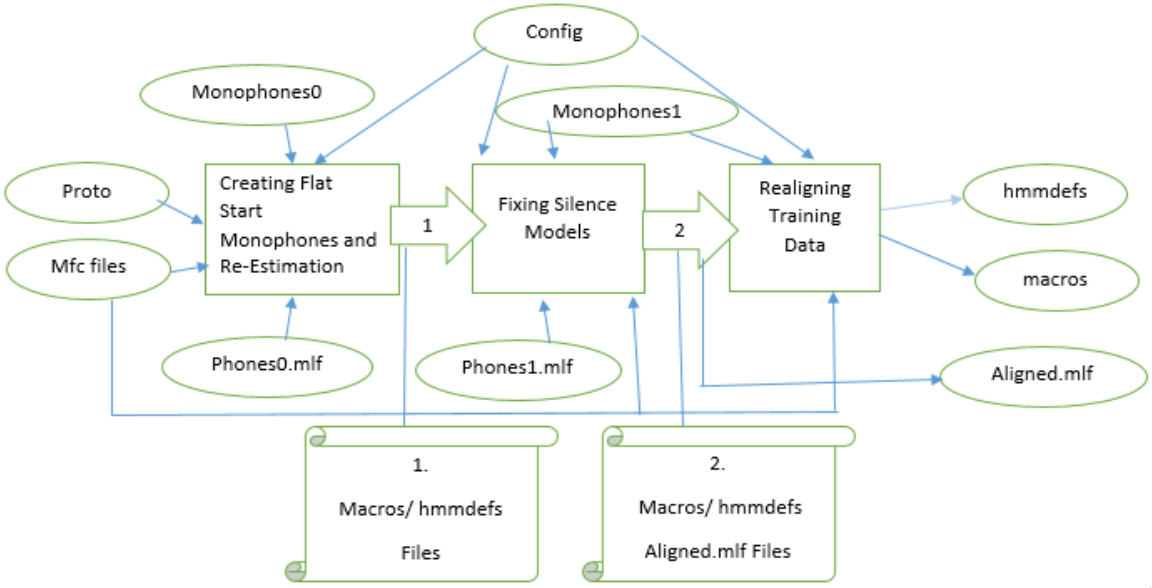



Figure 0.16 MonophonesHMM creation

### i) Creating Flat Start Monophones and Re-estimation

This is first sub-phase and at first proto file is to set manually, which defines model topology. The HCompV command of HTK scans a set of data files and results of the global mean and variance, then new version of HMM files are created. HCompV command also sets all the Gaussians in the resulted HMM which needs the re-estimation and so resulting HMM are re-estimated three more times by using HTK command "HERest" [22].

### ii) Fixing the Silence Models:

Fixing the Silence Models here means creating short pause (sp), which normally occurs in speech, between the words. This makes STT more noise tolerant in training phase and recognition phase. "sp" silence model are factually shorter than the "Sil" silence model (which occurs at the end of the sentence). HHed command is used to fix this.

### iii) Realigning the Training Data:

The dictionary can have multiple pronunciations for the same word. To select the best possible pronunciation, the HTK's HVite command is used to realign HMM after the re-

estimation of the results of the fixed Silence Models. Then save the results for retraining in a new transcript file, i.e., “aligned.mlf”[35]. After realigning, HMM again uses HERest command two more times which makes update in “aligned.mlf” and ”monophones1” files[22] as shown in Fig. 10.

#### 4.2.3.3 Tied-State Triphones HMM Creation

In a monophone Acoustic Model, Speech reorganize a monophone in the "context" of other monophones [30]. This can be done in two steps. Firstly, previously created monophone transcriptions are converted into triphone transcriptions and re-estimation the model is performed. Secondly, re-estimated triphone transcription are converted into tied-sate triphone transcriptions and again re-estimation is performed as shown in figure. Figure 4.19.

i) **Triphones Creation and Training:**

The triphone is the sequence of three monophones (group of three phones). It can be represented in the form of “L-X+R” where, “L” phone (left phone) proceeds “X” phone and “R” phone (right phone) follows it. For example, monophone and triphones of DIAL are given in Figure 4.18.

Monophone:” d ay l”

Triphone: “Sil d+ayd-ay+l ay-l”

Figure 0.17 Monophone and its triphone

The HTK command HLEd can be used to create triphones. It requires two files, aligned.mlf and a script file. And after creating the triphone files, HERest command is used to re-estimate the system with triphone.

ii) **Tied-State Triphones Creation and Training:**

After creating triphones, HHED command is used to tie states within triphone list which makes the STT more robust. It takes previously created HMM file with triphone file and creates a new version of HMM with tiedlist file which is used further for training of the

system. After that, re-estimate the system two more times with the help of HERest command.

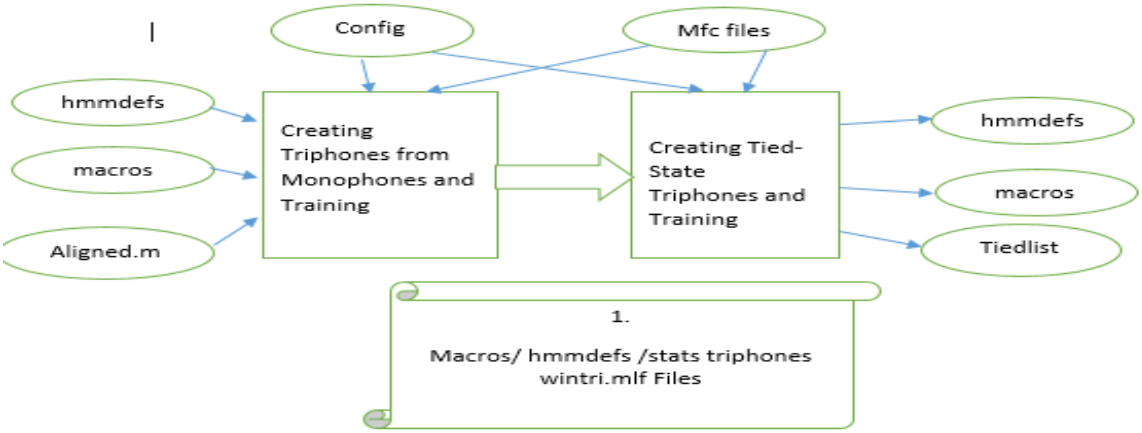



Figure 0.18 Tied-State Triphone Creation

### 4.2.3.4 Execution with Julius Interface:

"Julius" is the interface to decode and execute the Speech recognition system. Basically it requires groups of files: such as Grammar Definition, Acoustic Model (files monophone HMM files), triphone HMM files[41]. For Grammar definition, it requires .dfa, .dict files and for Acoustic model File, it requires lastly estimated Monophone HMM file and tiedlist for triphone HMM files which should be included in Julius configuration file as shown in Fi8. Julian command is required to execute the system. Command to execute system is sh g ow in Figure 4.20

```
dfa filename.dfa
-v      filename.voca
-h     hmm15/hmmdefs
-hlist tiedlist
-smpFreq 48000
```

Figure 0.1 Julian command for execution

# Chapter 5
# Testing and Result

In this chapter, the complete steps by steps implementation results have been discussed which will work in the background of application and will not appear to the user like how user's color image, Depth images, hand tracking is toot place.

## 5.1 Capturing the Depth Image:

The results of depth image as shown in the Figure 5.1 the depth of body the shows by the change of gray color intensity.

Figure 5.1 Depth Image

The object at same distance are with the same gray color intensity as shown in Figure 5.1.

## 5.2 Capturing the Body Frame:

The results of body frame as shown in figure 5.2

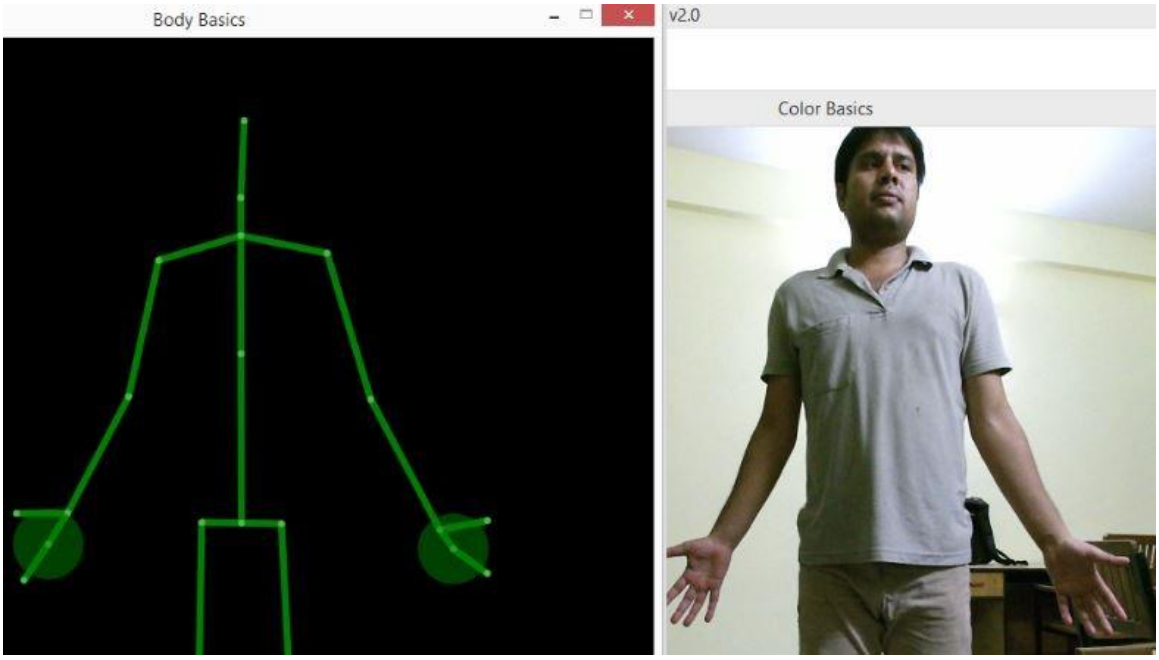



Figure 5.2 Body and joint image

## 5.3 Body joint tracking:

The Head joint and hand joint tacking is shown in Figure 5.2 .which is tracking by drawing red circle to the joints.

Figure 5.3 Head and hand joint tacking

## 5.4 Detecting the Hand state:

After tracking the hand, Hand state detection is also shown is Figure 5.3.


Figure 5.3 Hand tracking and Hand state detection

Kinect utility provides the track the hand their state as shown in Figure 5.3.

## 5.5 Touch-less NUI

Figure 5.4 set to touch less NUI

# Chapter 6
# CONCLUSION AND FUTURE WORK

## 6.1 Conclusion

We focused on finding the basic idea to develop touch less NUI using MS-Kinect HTK. We have shown that, even without the using of any external Machine learning algorithm, just using of Kinect SKD. One of the main objectives of Touch Less NUI is to provide natural, efficient, and flexible communication between the user and the computer. Human gestures including positions and movements of the fingers, hands, and arms represent one of the richest non-verbal communication modality, which together with the face expressions and head movements allow human users to interact naturally with the Computer. Thesis is wok on only one gesture. But Gestures can be static or dynamic,

This thesis proposed to understand and implement the touch less NUI using hand gesture recognition with Kinect SDK provides methods, which works under different lightning conditions Due to Kinect infrared vision capability efficiently.

The Touch less NUI will also provide a more efficient and natural interaction, modality for artistic applications. Medical applications will also significantly benefit from using visual hand gesture interaction for an efficient access to significant patient data during medical procedures in a sterile environment.

Another significant application will be in the sign language recognition for the deaf people. The Natural User Interface (NUI) proposed in this thesis uses the Kinect feature for hand gesture recognition.

# REFERENCES


[1]. JARRETT WEBB, JAMES ASHLEY. *"Beginning Kinect Programming with the Microsoft Kinect SDK"*. New York : Springer Science+ Business Media New York, 2012. ISBN-13: 978-1-4302-4104-

[2]. Tobii Gaze Interaction. [Online] TOBII. [Cited: 04 18, 2013.] "http://www.tobii.com/en/gaze-interaction/global/."

[3]. MICROSOFT. "*Teaching Kinect for Windows to Read Your Hands. Microsoft Research. [Online]"* 03 2013. [Cited: 04 17, 2013.] "http://research.microsoft.com/apps/video/dl.aspx?id=185502".

[4]. Skeletal Joint Smoothing White Paper. MSDN. [Online] MICROSOFT. [Cited: 04 26, 2013.]

[5]. Neural Network. Wikipedia. [Online] [Cited: 04 30, 2013.] http://en.wikipedia.org/wiki/Neural_networ

[6]. Natural User Interface: the Future is Already Here. Design float blog. [Online] [Cited: 04 17, 2013.] http://www.designfloat.com/blog/2013/01/09/naturaluser-interface/.

[7]. Kinect Coordinate Spaces. MSDN. [Online] MICROSOFT. [Cited: 04 25, 2013.] http://msdn.microsoft.com/en-us/library/hh973078.aspx.

[8]. Kinect for Windows Sensor Components and Specifications. MSDN. [Online] MICROSOFT. [Cited: 04 30, 2013.] http://msdn.microsoft.com/enus/library/jj131033.aspx.

[9]. Jelinek F., Bahl L.R., Mercer R.L., "*Design of a linguistic statistical decoder for the recognition of continuous speech*", IEEE Transactions on Information Theory, vol. 21(3), 1975, pp. 250-256.

[10]. MICROSOFT. Human Interface Guidelines v1.7.0. [PDF] 2013.

[11]. Kemble K. A., "*An Introduction to Speech Recognition",* Program Manager, Voice Systems Middleware Education IBM Corporation, unpublished.

[12]. Furui S., "Speaker independent isolated word recognition using dynamic features of speech spectrum", IEEE Transactions of Acoustics, Speech, Signal Processing, ASSP, vol. 34(1), February 1986, pp. 52-59.

[13]. Hasanabadi H., Rowhanimanesh A., Yazdi H. Tabatabaee, Sharif N., *"A Simple and Robust Persian Speech Recognition System and Its Application to Robotics*", in Proc. International Conference on Advanced Computer Theory and Engineering, Phuket, Thailand, 20-22 December, 2008.

[14]. MICROSOFT. Kinect for Windows SDK v2. Known Issues. MSDN. [Online] MICROSOFT. [Cited: 04 30, 2014.] http://msdn.microsoft.com/enus/library/dn188692.aspx.

[15]. Raza A. A., Hussain S., Sarfraz H., Ullah I., Sarfraz Z., "*Design and development of phonetically rich Urdu speech corpus"*, in Proc. Speech Database and Assessments, Oriental COCOSDA International Conference, Beijing, China, 2009, pp. 38-43

[16]. Furui S., "*Speaker independent isolated word recognition using dynamic features of speech spectrum*", IEEE Transactions of Acoustics, Speech, Signal Processing, ASSP, vol. 34(1), February 1986, pp. 52-59.

[17]. Sign Language Recognition with Kinect. [Online] [Cited: 04 18, 2013.] http://page.mi.fu-berlin.de/block/abschlussarbeiten/Bachelor-Lang.pdf

[18]. Itakura F., "*Minimum prediction residual applied to speech recognition*", J. of ASSP, vol. 23 (1), 1975, pp. 67-72.

[19]. Raza A. A., Hussain S., Sarfraz H., Ullah I., Sarfraz Z., "*Design and development of phonetically rich Urdu speech corpus*", in Proc. Speech Database and Assessments, Oriental COCOSDA International Conference, Beijing, China, 2009, pp. 38-43.

[20]. Sakoe H., Chiba S., "*Dynamic programming algorithm optimization for spoken word recognition"*, J. of ASSP, vol. 26 (1), 1978, pp. 43-49

[21]. Liu Y., Shriberg E., Stolcke A., Peskin B., Ang J., Hillard D., Ostendorf M., Tomalin M., Woodland P., Harper M., *"Structural metadata research in the EARS program"*, in Proc. ICASSP, vol. 957, 2005, pp. 1-4.

[22]. Martin T. B., Nelson A.L., Zadell H. J., "*Speech recognition by feature abstraction techniques"*, August 1964.

[23]. Myers C. S., Rabiner L. R., *"A level building dynamic time warping algorithm for connected word recognition*", IEEE Transactions on Acoustics, Speech, Signal Processing, ASSP, vol. 29(2), 1981, pp. 284-297.

[24]. Nagata K., Yasuo K., Seibi C., "*Spoken digit recognizer for Japanese language"*, J. of AES, vol. 12(4), October 1964, pp. 336-342.

[25]. Olson H. F., Belar H., *"Phonetic typewriter*", J. of Acoustical Society of America, vol. 28 (6), 1956, pp. 1072-1081.

[26]. Pavlovic N., "*Natural Language Processing and Speech Enabled Applications*", Computer Science Department City Liberal Studies, Affiliated Institution of University of Sheffield.

[27]. Peiiagarikano M., Bordel G., *"Speech-to-Text Translation by a non-word Lexical based system*", in Proc. Fifth International Symposium on Signal Processing and its Applications, ISSPA '99, Brisbane, Australia, August, 1999, pp. 111-114.

[28]. Rabiner L.R., "A tutorial on hidden Markov models and selected applications in speech recognition", in Proc. IEEE, vol. 77 (2), February 1989, pp. 257-286.

[29]. Rabiner L. R., Levinson S. E., Rosenberg A. E., Wilpon J.G., "*Speaker independent recognition of isolated words using clustering techniques",* IEEE Transactions on Acoustics, Speech, Signal Processing, ASSP, vol. 27(4), 1979, pp. 336-349.

[30]. Reddy D. R., "An approach to computer speech recognition by direct analysis of the speech wave", Computer Science Dept., Stanford Univ., 1966.

[31]. Sakai T., Doshita S., "The phonetic typewriter information processing", in Proc. IFIP Congress, Munich, Germany, 27 August - 1 September, 1962.

[32]. Sakoe H., Chiba S., "Dynamic programming algorithm optimization for spoken word recognition", J. of ASSP, vol. 26 (1), 1978, pp. 43-49.

[33]. Shinoda K., Lee C. H., "A structural Bayes approach to speaker adaptation", IEEE Transactions of Speech and Audio Processing, vol. 9(3), 2001, pp. 276-287.

[34]. Sivaraman G., Samudravijaya K, "Hindi Speech Recognition and Online Speaker Adaptation", in Proc. International Conference on Technology Systems and Management, ICTSM, Czech Technical University in Prague, Czech Republic, 2011, pp. 27-30.

[35]. Suzuki J., Nakata K., "Recognition of Japanese vowels - preliminary to the recognition of speech", J. of Radio Research Lab, vol. 37 (8), 1961, pp. 193-212.

[36]. Tomalin M., Diehl F., Gales M.J.F., Park J., Woodland P.C., "Recent Improvements to the Cambridge Arabic Speech to Text Systems", in Proc. ICASSP, Dallas, Texas, USA, 2010, pp. 4382-4385.

[37]. Tutorial: Create Acoustic Model – Manually, [online], Available : http://www.voxforge.org/home/dev/acousticmodels/windows/create/htkjulius/tutorial, [Accessed: 15 December 2011].

[38]. Varga A. P., Moore R. K., "Hidden Markov model decomposition of speech and noise", in Proc. ICASSP, vol. 2, 1990, pp. 845-848.

[39]. Viterbi A. J., "Error bounds for convolutional codes and an asymptotically optimal decoding algorithm", IEEE Transactions on Information Theory, vol. 13(2), April 1967, pp. 260-269.

[40]. Vintsyuk T. K., "Speech discrimination by dynamic programming", J. of Kibernetika, vol. 4 (2), 1968, pp. 81-88.

[41]. What can I do with Speech Recognition, [online], Available: http://windows.microsoft.com/en-US/windows7/What-can-I-do-with-Speech-Recognition, [Accessed : March 15, 2012].