Gesture and Voice Control of Internet of Things

A thesis presented in partial fulfilment of the requirements for the

degree of

Master of Engineering

in

Electronics and Computer Engineering

at Massey University, Auckland,

New Zealand.

Xiao Han

October 2015

ACKNOWLEDGEMENTS

I would like to thank my supervisor Dr Mohammad Rashid for his guidance throughout my master study. His enthusiasm, encouragement, and faith in me have been extremely helpful.

Next, I would like to thank Dr. Liqiong Tang who is the one inspired me and helped me find the direction of study. Her support and faith on me have been very meaningful and made me start to believe myself too.

I would like to thank all my friends, my sister and my parents for their companionship and support.

ABSTRACT

Nowadays, people's life has been remarkably changed with various intelligent devices which can provide more and more convenient communication with people and with each other. Gesture and voice control are becoming more and more important and widely used. People feel the control system humanized and individualised using biological control.

In this thesis, an approach of combined voice and gesture control of Internet of Things is proposed. A prototype is built to show the accuracy and practicality of the system. A Cortex-A8 processor (S5PV210) is used and the embedded Linux version 3.0.8 has been cross-compiled. Qt 4.8.5 has been ported as a UI (User Interface) framework and OpenCV 2.4.5 employed as vision processing library. Two ZigBee modules are used to provide wireless communication for device control.

The system is divided into control station and appliance station. The control station includes development board, USB camera, voice recognition module, LCD screen and ZigBee module. This station is responsible for receiving input signal (from camera or microphone), analyzing the signal and sending control signal to appliance station. The appliance station consists of relay, ZigBee module and appliances. The ZigBee module in the appliance station is to receive control signal and send digital signal to connected relay. The appliance station is a modular unit that can be expanded for multiple appliances.

The system can detect and keep tracking user's hand. After recognizing user's gesture, it can control appliances based on certain gestures. Voice control is included as an additional control approach and voice commands can be adjusted for different devices.

TABLE OF CONTENTS

LIST OF ABBREVIATIONS

Various specialized abbreviations are used in this thesis as listed below:

API……Application Programming Interface

CamShift……Continuously Adaptive Mean Shift

CAN……Controller Area Network

CMOS……Complementary Metal-oxide Semiconductor

COM……Common

Cramfs……Compressed ROM File System

CPU……Central Processing Unit

DNN……Deep Neural Networks

DTW……Dynamic Time Warping

EEG……Electroencephalograms

EN……Enable

EPIC……Electric Potential Integrated Circuit

GPIO……General-Purpose Input/Output

GND……Ground

GUI……Graphical User Interface

HMM……Hidden Markov Model

HSV……Hue-Saturation-Value

IoT……Internet of Things

IR……Infrared

ISO……International Organization for Standardization

JFFS……Journaling Flash File System

LCD……Liquid-Crystal Display

LED……Light-Emitting Diode

MCU……Microcontroller Unit

MFC……Mel Frequency Cepstum

MFCC……Mel Frequency Cepstum Coefficients

MSEPF……Mean Shift Embedded Partial Filter

NFS……..Network File System

NC……Normally Closed

NO……Normally Open

RAM……Random-Access Memory

RFID……Radio-Frequency Identification

RGB……Red-Green-Blue Color Model

ROM……Read-Only Memory

SASOM……structure adaptive self-organizing map

SCFG……Stochastic Context Free Grammar

SIFT……Scale Invariant Feature Transform

SPDT……single-pole, double-throw

SURF……Speeded Up Robust Features

TTL……Transistor–transistor Logic

UART……Universal Asynchronous Receiver/Transmitter

UI……User Interface

URC……Universal Remote Console

USB……...Universal Serial Bus

YAFFS……Yet Another Flash File System

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1 INTRODUCTION

1.1 Background

1.1.1 The Rise of Internet of Things

The Internet of Things (IOT) is a vision and a concept coined by Kevin Ashton [1] in the early 2000's while working at the MIT's AutoID lab. In the early years of IoT, RFID (Radio Frequency Identification) and sensor technologies were the focus as Kevin Ashton writes, "If we had computers that knew everything there was to know about things—using data they gathered without any help from us -- we would be able to track and count everything, and greatly reduce waste, loss and cost. We would know when things needed replacing, repairing or recalling, and whether they were fresh or past their best. We need to empower computers with their own means of gathering information, so they can see, hear and smell the world for themselves, in all its random glory. RFID and sensor technology enable computers to observe, identify and understand the world—without the limitations of human-entered data." The concept has grown enormously during the last one and half decade. Today IoT describes a system where Internet-enabled "things" in the physical world, containing sensors, actuators and capable of interacting with the environment, are connected to the Internet via wired and wireless Internet connections.

There are multiple enabling technologies which play important roles in the advancement of IoT. The technologies can be divided into two groups:

i) Connectivity and Networks: the Internet Protocol version 6 is created to replace the old version. It has very large address space which can provide much more addresses for connecting millions even billions of devices. Connected devices need networks to exchange information with each other. There are many kinds of wireless network technologies. For example, ZigBee as a low-cost, low-power wireless technology is often used in IoT and M2M communication. It enables data

transmission through mesh networks and is suitable for smart home, embedded sensing and building automation [2]. Compared to ZigBee, Wi-Fi has faster data transfer speed and is optimized for large data transfer. But it is not a power-efficient network which makes it unsuitable for many M2M applications [2].

ii) Sensors and Microprocessors: the advancement of sensors and microprocessors includes four aspects:

1. 1. Smaller and more durable sensors: With the improvement of manufacturing and technologies, new sensors can reach very small dimensions to fulfill diverse objectives. More durable sensors are applied for distinctive environments and specific purposes, such as for out-space extreme low temperature use.
2. 2. Diversity of processor chips: The manufacturers keep optimizing their chips in different ways - to be more powerful or dedicated. Such as Xeon processors [3] from Intel provide very powerful computing ability while Qualcomm provides a chip for smart-watch which is capable for Android Wear OS with a very low power consumption and Bluetooth Low Energy connection [4].
3. 3. Improvement of processor performance: The performance of processors is improving due to the innovation and updating of the design and micro-architecture of processor. Such as the stepping micro-architectures evolution of Intel from 8086, Pentium to Core and Skylake [5].
4. 4. Lower costs of electronic components: The price of electronic components is rapidly decreasing due to the improvement of manufacturing technology.

As Figure 1.1 shows, from smart watches to televisions, from mobile phones to security cameras, "things" in IoT can be regarded as all the objects that can connect into networks, collect or exchange data. With the rapidly increasing number of connectable objects, the IoT is now all around in our daily lives. The growth of the IoT has influence on everyone and everything, and will make great differences in the next decades [6]. With the help of the IoTs, the efficiency of

information exchange, safety of daily life and the convenience of communication are improved.

The biggest benefits of the IoTs are in five areas:

1. Body: Wearable devices will be essential in people's life which can help them to collect information of daily exercise, health level and fitness [7]. There are some other functions, such as to monitor children's activities or to keep informed of someone's location.
2. Home: Sensors and smart appliances will be widely used and people can monitor and control almost everything remotely. For instance, to control heater or air condition and to get alarm information from home security system.
3. Community: Large embedded systems will be able to improve public transportation and power supply. Information about problems and data can be easier to collect and analyze.
4. Goods and services: More sensors will be used in manufacturing companies which can keep tracking and collecting information about goods. This can greatly improve the productivity of goods supply and distribution.
5. Environment: Real-time information can be given using devices and readers which can enable closer monitoring of forests, ground, water and air.

Figure 1.1: The Internet of Things [6]

1.1.2 Gesture and Voice Control on IoT

Gesture and voice recognition are gaining importance as the new interfaces for the Internet of Things. There are attempts to develop applications like using gesture to control smart homes and voice command to communicate with electronic devices. They will be more widespread in the future. This trend is based on the growth of accuracy and efficiency of gesture recognition and voice synthesis [8].

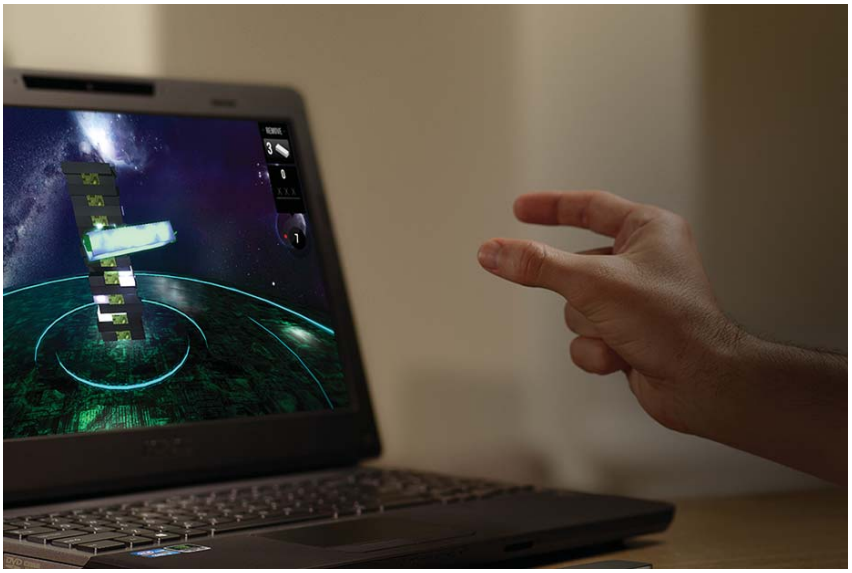

Figure 1.2: An example of gesture control [9]

Gesture-based Systems should meet the following criteria:

1. Specifiable: Gesture should be specifiable which means the gesture should be able to ported to the system. Only the system is able to recognize a specific gesture, can the gesture be meaningful. Figure 1.2 shows an example of gesture control.

2. Accuracy: Accuracy is affected by the amount of gesture commands, the complexity of gesture and the analysis time. Generally, three-dimensional gesture system tends to be more accurate.

3. Efficient: The speed of recognition is very important. Only an interface with high speed of recognition and feedback would be easy to operate and accepted by the users.

4. Ability of training: An ideal system is able to allow user to adjust or classify gestures. There should be a training system which can be easy to manipulate and takes not much time to accomplish.

Using voice recognition to control the Internet of Things is quite popular nowadays. People are getting used to using voice commands to control their phones and intelligent watches, even to input words instead of typing. But there are still many features need to be improved. One of the biggest drawback is the accuracy might be very low when the voice recognition system is not familiar with the user's accent.

1.2 Objectives

The objectives of this thesis are:

1. To use gesture recognition as a control method based on the embedded system. Recognize gestures according to the real-time images which are captured by the camera. The recognition should meet basic requirements for accuracy and efficiency.

2. To apply voice recognition to control the Internet of things. Use voice recognition module to recognize certain voice commands and control the relevant devices. The voice commands should be able to be adjusted for different situations.

3. To build a prototype of the Internet of things. The prototype should be capable to display how to use gesture and voice recognition to control several target devices. The gesture and voice commands used in the prototype should be able to be trained by the user.

4. To develop an user interface with the ability to add more control panel. The user interface needs to be user-friendly. Each control panel is for the unique device. It should be flexible to add more control panels for new devices.

5. To test the accuracy and efficiency of the system. Perform tests for both gesture and voice recognition under different practical environments.

1.3 Organization of the thesis

The thesis consists of 6 chapters. Chapter 2 presents an overview of the Internet of things. Chapter 3 gives a literature review while we present the system design and implementation in Chapter 4. In Chapter 5, we evaluate the system performance with two user tests. Finally in Chapter 6, we conclude the thesis and provide suggestions for future direction.

CHAPTER 2 AN OVERVIEW OF THE INTERNET OF THINGS

2.1 Introduction

Numbers of breakthroughs of technology have together enabled the advance of the Internet of Things. These breakthroughs include the improvement of sensors, bandwidth, processing, smart phones, ubiquitous wireless coverage and big data analytics. There are many successful products like wearable devices which have already become very common in daily life. There are more internet-connected devices than human now, and the number is expected to increase greatly in the next decade.

2.2 The History of IoT

The Internet of Things is considered as the new revolution of the Internet. Kevin Ashton, the visionary of the IoT, indicated that IoT could connect all devices into the network from daily used devices to industrial equipment. And the possibility to collect all information and manage all devices in one Internet can greatly improve the efficiency and convenience.

The early concept of the Internet of Things is to manage and inventory objects by using radio-frequency identification (RFID). One of the examples is the "Electronic Product Code" which is an idea of object identification scheme using Internet and RFID technology [10].

The concept of the Internet of Things is also based on another idea from Machine-to-Machine (M2M) applications - "a machine has more value when it is networked and that the network becomes more valuable as more machines are connected" [11]. In the very beginning, the Internet was a network for computers connecting together to share data. With the improvement and convergence of technologies, such as wireless communication, sensors and embedded systems,

the types of "computers" that can connect into the Internet have been significantly expanded.

Nowadays, the collection of applications, machines and all the things connected together to share information is the Internet of Things.

To connect all devices into the Internet first needs millions even billions of addresses. IPv6 which is intended to replace the older version IPv4 was created to provide about 3.4×10^{38} addresses.

Once connected, networks are required to enable devices to exchange information with each other. There are several wireless network technologies such as ZigBee, NFC (near field communication) and Wi-Fi. Developers can choose different technologies for the particular environment according to their different advantages and disadvantages.

ZigBee is a wireless technology which can transfer data through mesh networks. Its bandwidth and transfer rates are much lower than Wi-Fi, but it is ideal for applications that demand for low power and low cost. NFC is quite different from ZigBee and Wi-Fi. As its effective range is very small, it can only be used in some special cases like contactless payment and card access. The typical Wi-Fi technology would be another choice of IoT network as its reasonable coverage area and fast transfer speed, and the only drawback is the power consumption. To solve the power consumption issue for Wi-Fi, Qualcomm has unveiled its new technology of Wi-Fi platform for IoT which provides lower power management techniques [12].

Apart from the connection technologies, sensors and processors are the other important elements of IoT. New technologies of sensors and microprocessors enable new sorts of IoT applications and workflows.

The sensors are becoming smaller which makes the wearable device a reality while more durable which expands the usefulness. Sensors in different sizes can be implemented into various applications. Meanwhile more durable sensors are

available for different environments, even for inner engine under high temperature and pressure. With the addition of new sensors, IoT network would be abundant with more information.

For the processors, on one hand, the manufacturers are adding more cores or processors into one chip to enhance the performance; on the other hand, the innovation of the design and micro-architecture of processors keeps carrying out by manufacturers to provide higher efficiency. An outstanding example is the SyNAPSE chip from IBM which is an electronic neuromorphic machine technology providing very powerful computing ability [13]. The improvement of processor performance provides the ability to complete complicated targets for IoT systems. Apart from the performance, dedicated processors that aimed at certain target help the improvement of IoT as well. Bluetooth Smart network processors from STM are dedicated processors which provide Bluetooth solutions with ultra-lower power consumption[14].

2.3 The Landscape of IoT

In the future, five key area will become the most attractive and promising, shown in Figure 2.1 [7].

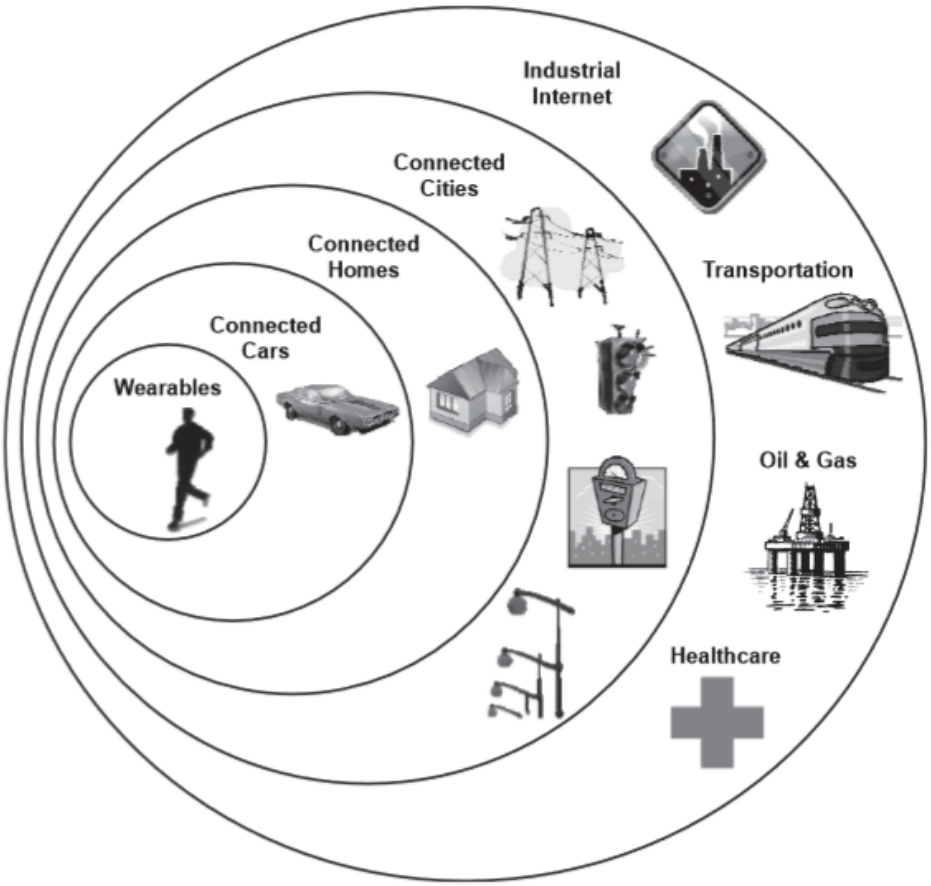

Figure 2.1: The landscape for IoT [7]

Those five key area are divided by their relationships with people:

1. The closest one is wearable device which mainly gathers information from body and provides certain resources and feedbacks.
2. The second circle is connected car. The embedded systems in cars provide the possibility of Internet activities, such as automatic crash notification, navigation and even car-to-car communication.
3. The third circle of this landscape is connected home, which can also be called as home automation. It may include the control of lighting, entry, secure, appliances and all other domestic systems.

4. The fourth circle is connected cities, which can be treated as the expansion of the connected homes. The city's lighting, energy and all services can be connected in the network for the idea of Smart City [15].
5. The biggest circle of the landscape is the industrial internet, which combines multi-field technologies to analyze data that got from machines and control operations.

2.4 IoT Components

In IoT network, there are mainly 3 different types of components: connectivity, control and sensor.

1. As the IoTs always need to connect with the real internet or other devices, the connectivity components are necessary and significantly important. Wired connection or wireless connection could be selected or combined to use in certain IoT network based on connection distance and speed requirements. Typical wireless module and technologies are Wi-Fi and ZigBee.
2. Control type of components is much more variant depending on different targets. The main part of the controlling components is the processor. There are many kinds of processors for different applications, such as ultra low power Microcontroller Units (MCU) for low system energy requirement which can easily be found in the market. For example, MSP430 MCU which is invented by Texas Instruments is one of the lowest power microcontroller platform [16]. With the well-tuned platform, the ultra low power MCU could keep very low power consumptions in both active(as low as 100 μA/MHz) and sleep modes (as low as 360 nA/MHz) and can still provide a reasonable wakeup time. In comparison, PIC16LF1509 as another popular ultra low power MCU, has even lower active power consumption: as low as 30 μA/MHz [16]. But its sleep power consumption is higher than MSP430.
3. To detect certain physical properties and transfer them into digital or analog signals, sensors are the most vital components in IoT solutions. There are many types of sensors: pressure sensor, temperature sensor, ultrasonic sensor,

humidity sensor and displacement sensor. The improvement of sensor has brought huge changes to IoT.

2.5 Technologies in IoT

The Internet of Things continues to develop with a huge numbers of related technologies such as Cloud computing, Big Data, robotics and communication technologies [17]. Voice recognition and gesture control are two of the popular communication technologies in Internet of Things.

2.5.1 Voice Control in IoT

Voice recognition is widely used in IoT devices and there are plenty of voice recognition apps. As the limitation in size and using situation, the typical input method cannot be performed, thus voice control is the best substitution. There is increasing interest in using human voice to interact with computing devices. Forrester says "voice control will be the next battleground" for the technology companies [18].

There are many voice recognition methods and strategies in the field for acoustic modelling and language modelling, such as Dynamic time warping (DTW), Deep Neural Networks (DNN) and Hidden Markov Models (HMMs). Nowadays, Most of voice recognition systems are based on Hidden Markov Models which is easy to be trained and simple to use.

Voice recognition can be widely used for lots of applications in Internet of Things. One of the common usage is for in-car systems. Simple voice commands can be used to make phone calls, switch radio or music and set navigation.

Almost all of the big-name companies have provided their voice recognition software. Take mobile phone's voice command system as an example. It allows the users to make command just by talking to it. It can help users sending messages, recording events and making calls. Voice control is the key technology for wearable devices because most of the wearable devices take voice as the main

input approach. Some daily used wearable devices like smart watch, can collect information and organize it, like receiving messages, updating weather information and tracking user's fitness [19].

2.5.2 Gesture Control in IoT

Gesture control is becoming mainstream nowadays. The success of some popular gesture-control-based devices has made this area even more competitive.

Xbox Kinect is presently the most influential 3D camera which can recognize body and hand gestures and bring a novel development of the interaction between human and computer [20]. There are many applications using Kinect as a depth camera to develop gesture recognition system which have very high accuracy. Figure 2.2 shows a gesture control system using Leap Motion which can displace traditional mouse and let users to control computers with finger or hand movement [21].

Figure 2.2: Display of Leap Motion [21]

Internet of Things is getting more and more important in our daily life and researching area. Both gesture and voice control are used as popular control methods in the Internet of Things. In this chapter, the history, landscape,

components and technologies of IoT are introduced. In the next chapter, the related literature of gesture and voice recognition are discussed.

CHAPTER 3 LITERATURE REVIEW AND RELATED WORKS

Gesture and voice recognition are widely used for human computer interaction. There are plenty of gesture and voice control applications using different technologies .In this chapter, the approaches of hand gesture recognition are discussed. The classification, basic process and algorithm of voice recognition are introduced. Five related works are critically reviewed.

3.1 Hand Gesture Recognition

There are two types of gesture recognition approaches: vision-based approaches and non-vision-based approaches.

i) Non-vision base approach: The typical case using this kind of approach is the data glove which can detect finger motion through sensors and then transfer it into electrical signals for recognition. This approach is often used in wearable devices or ported into smart phones and tablets [22].

ii) Vision based approach: Cameras and different kinds of methods for image processing are involved in this approach. An examples to use Kinect which is quite popular nowadays. Kinect uses RGB camera, infrared emitter and infrared CMOS camera to measure the depth of image.

3.1.1 Non-Vision Based Approaches

Based on the technology used to detect hand gesture, non-vision based approach is divided into two types:

1. system using non-contact sensors.
2. system using contact sensors.

3.1.1.1 System using non-contact sensors

There are various non-contact sensors in the market. Four widely used sensors are introduced:

(a) Ultrasonic sensing system

Ultrasonic sensing technology can detect the motion of gestures in various lighting and noise environments. Using ultrasonic sensing device can calculate the distance between the object and the device based on the reflected ultrasonic signal. When the user making a gesture within the effective range, the locations of hand during the period can be measured.

Liu et al. [23] used ultrasound to perform 3D gesture recognition on mobile devices. The prototype of this project is shown in Figure 3.1. The ultrasonic sensor they used is MB1010 rangefinder [24]. It is a 42kHz ultrasonic sensor with the maximum range of 645cm. There are 4 sensors as the arrows indicate in Figure 3.1. They are working simultaneously to collect 4 distance data. Then according to 4 distance data, a six degree-of-freedom gesture is presented in 3D format. This device supports 12 kinds of gesture recognition which is quite enough for normal operations on mobile devices.

Figure 3.1: The prototype of SoundSense system. The four arrow indicates four ultrasonic sensors [23]

(b) Infrared sensing system

There are two main methods for gesture recognition using infrared sensor:

1. Position-based method: According to the data collected, the position of the object can be estimated. The timing of the movement of the position data is checked to see if any gestures occurred [25]. The process involves three stages: 1) to convert raw data to distance data; 2) to estimate the position according to distance data; 3) to time gestures.

2. Phase-base method: Compared with position-base method, there is no location data calculated. As Figure 3.2 shows, the gesture is recognized only based on the analysis of LED's feedback. When the hand is right above the LED, the feedback will achieve the maximum value. The Si114x sensor is a proximity sensor [26]. It is used with three LEDs to detect gesture. If the hand is moving above the LED D2 and D3, the moving direction can be determined based on the timing of the increase of feedback values.

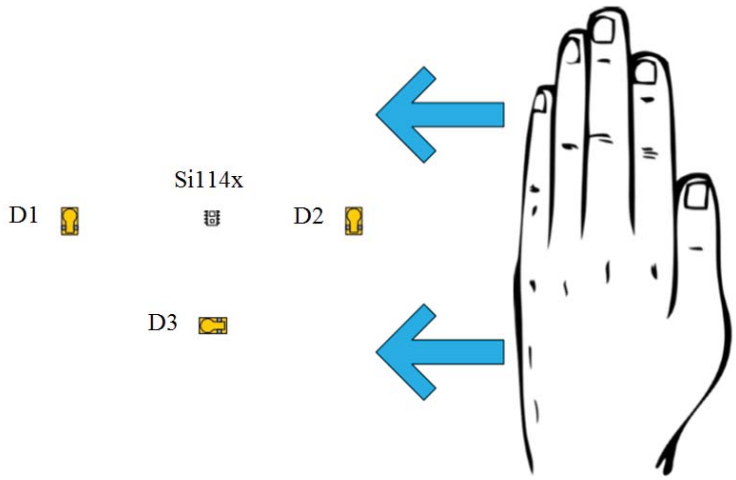

Figure 3.2: An example of 3 LED system [25]

(c) Electrical field sensing system

This kind of system can recognize gestures by sensing the feedback of electric field. Kim et al. [27] used EPIC (Electric Potential Integrated Circuit) sensors to measure the change of electrical system. EPIC sensor is an AC coupled device

which is able to measure the electric potential. And using multiple EPIC sensors can get differential data which represents the surrounding electrical field. Because human body can cause detectable change in the electrical field, when different gestures occurs, a corresponding type of perturbation of the local electrical field will appear. Kim used four sensors to get two differential data. According to the differential data, a certain type of gesture can be recognized. Figure 3.3 shows a typical internal circuit of a EPIC sensor. The sensor is capacitive coupling, so for a certain application, the size of the probe electrode is very important due to the strong corresponding to the input capacitance [28].

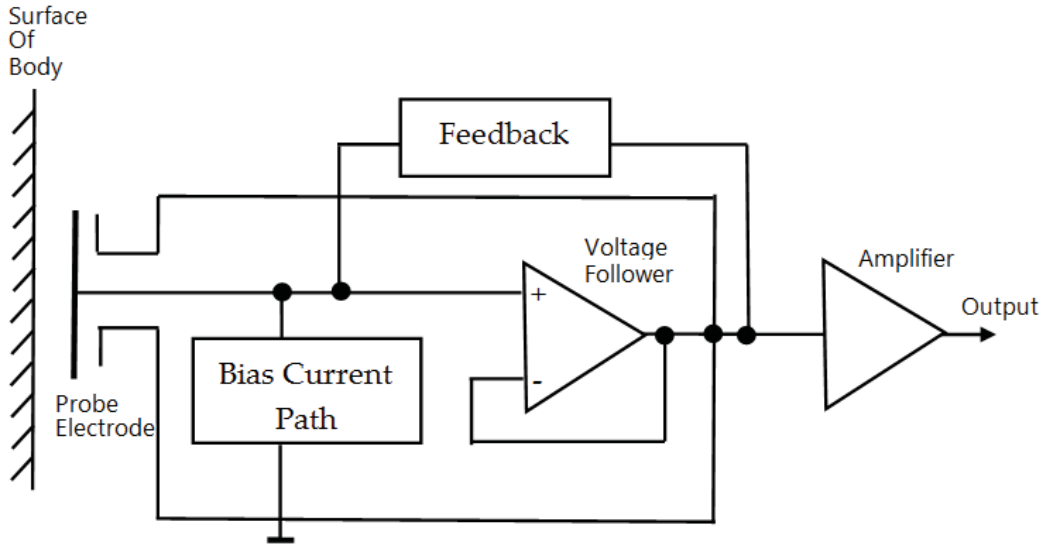

Figure 3.3: An internal circuit of a EPIC sensor [28]

(d) Radio frequency system

Radio frequency system is similar as ultrasonic sensing system because they are both based on the measurement of frequency. There are plenty of cases for radio frequency system. Kellogg et al. [29] developed an ultra-low power gesture recognition system called AllSee which uses RFID tags and is able to recognize eight hand gestures at a high accuracy. The prototype of AllSee is shown in Figure 3.4. It can be considered as two parts: a receiver and a microcontroller. The receiver is pluggable and responsible for receiving the amplitude of RFID signals.

The microcontroller used in the device is MSP430 which has UART interface and LEDs.

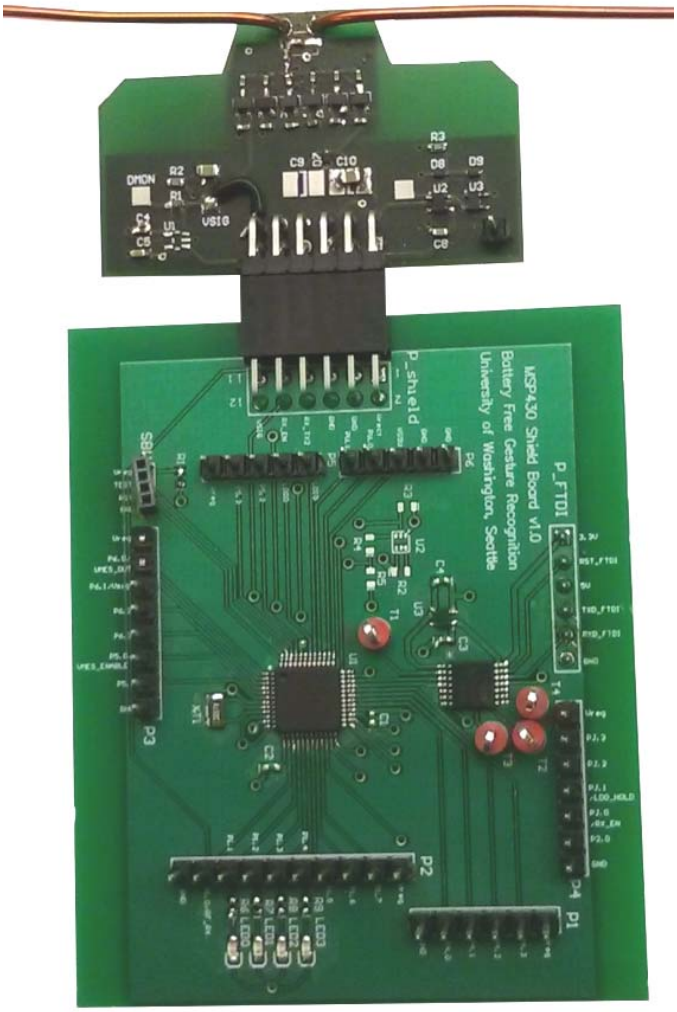

Figure 3.4: The prototype of AllSee [29]

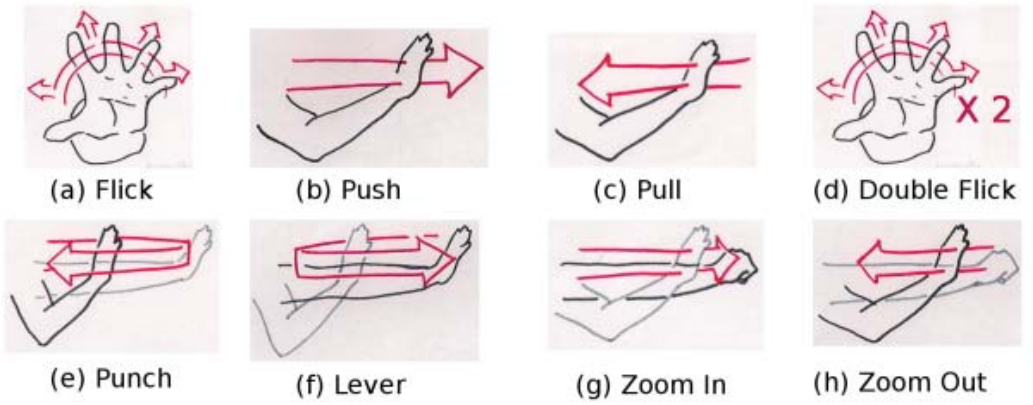

Figure 3.5: Eight Gestures AllSee can detect [29]

AllSee can detect eight different gestures as shown in Figure3.5. Figure 3.6shows the corresponding change of the output from the detector. Every gesture has a unique change.

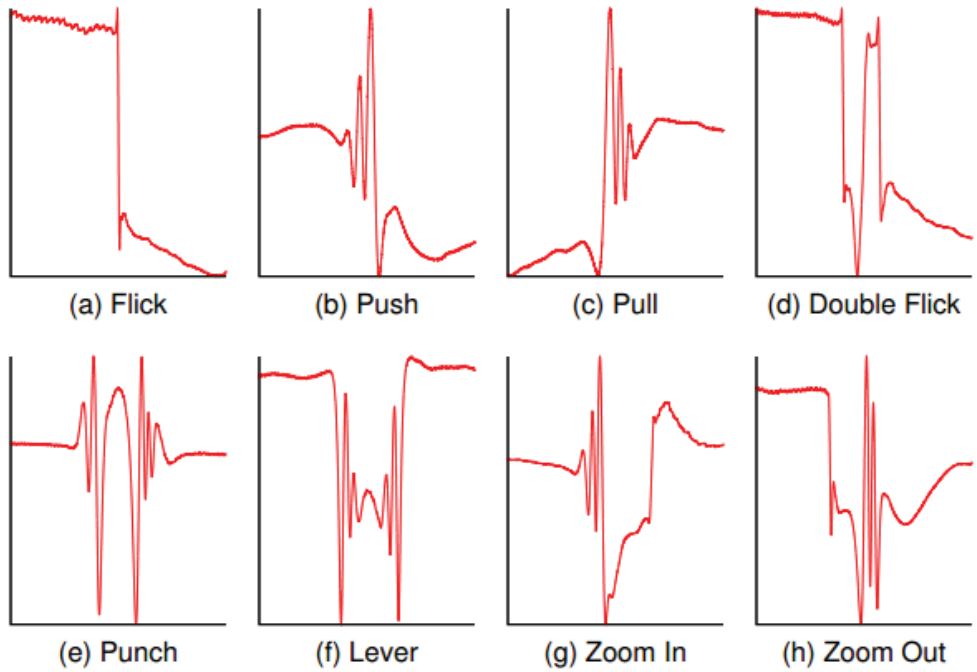

Figure 3.6: The output of the detector when the using making a certain gesture [29]

3.1.1.2 System using contact sensors

Contact sensors are usually employed in wearable devices. There are four common types of sensors:

(a) Accelerometer sensor system

Using accelerometer sensor is a suitable method for ubiquitous gesture-based interactions. J. Wu et al. [30] used WiiMote to develop a frame-based gesture descriptor for 3D gesture recognition. Figure 3.7 shows the WiiMote controller and the accelerometer axes. The three axes are represented as different colored lines. Users need to hold WiiMote and make gestures, and the system can recognize the gesture according to the acceleration data. The accelerometer sensor used in WiiMote is ADXL330 [30]. The other import device employed in WiiMote is an optical sensor which can determine the direction that WiiMote points.

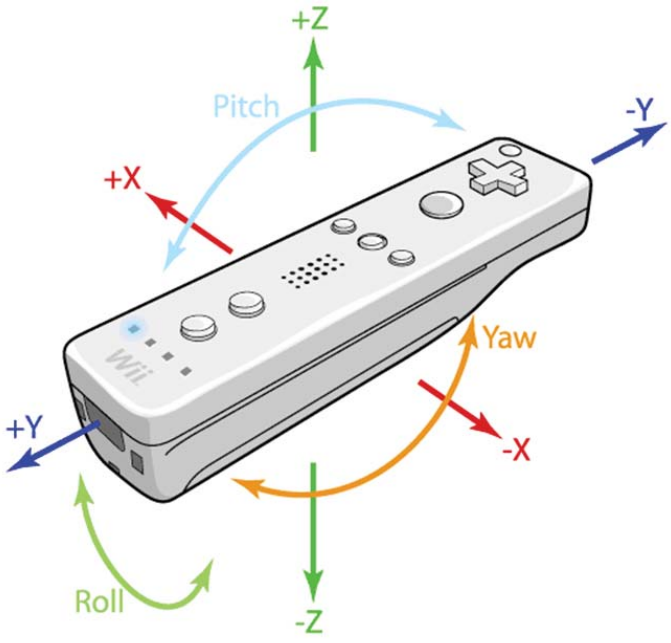

Figure 3.7: WiiMote controller and the accelerometer axes [30]

(b) Gyroscope sensor system

Theoretically, gyroscope-based gesture recognition can achieve high accuracy with very low cost. The gyroscope sensor can measure the rotational velocities. It

is widely used in camera-shake correction, car navigation and video games. A Gyro mouse is shown in Figure 3.8. Normally, the Gyro mouse is applied as a 3D mouse or pointer which allows users to control their computer/laptop from as far as 10 meters away. There are six function buttons including the left/right mouse buttons, scroll wheel and three programmable mouse buttons.

Andreas Hofer et al. [32] used gyro mouse and machine learning method to perform gesture recognition. The Gyro sensor is used as an input device to get rotational velocities. When the user holds the Gyro sensor and moves it, the sensor measures rotation around two axes: x and y, as indicated in Figure 3.8.

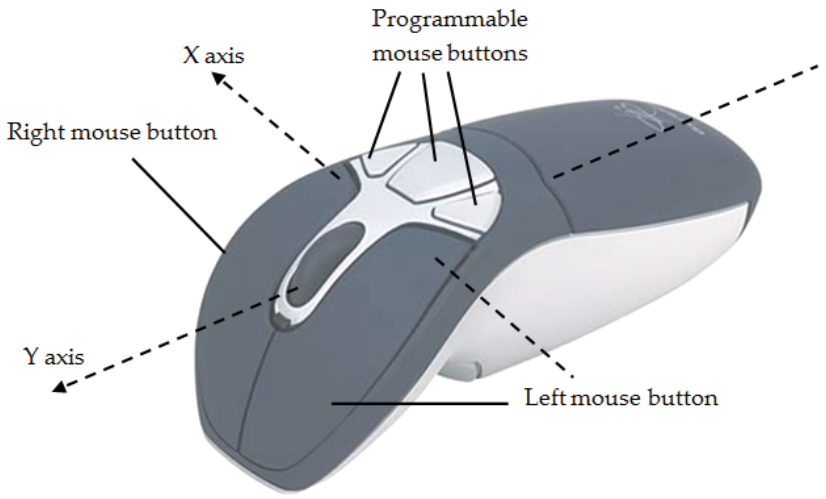

Figure 3.8: Gyro sensor [32]

(c) Magnetic sensor system

Using magnetic sensor to recognize gesture can accomplish non-touch control within the 3D space around a smart phone or tablet. H. Ketabdar et al. [33] introduced a new interaction using magnetic field sensor for gesture recognition. In this project, the user needs to hold a suitable shaped magnet to draw gestures around the target device. The compass sensor which is ported into the device will be affected according to the gesture. Based on the change of magnitude signal, the system can determine which gesture has been performed and make the corresponding command. Figure 3.9 shows the user using a ring shape and a pen shaped magnet to control a smart phone.

Figure 3.9: Using magnet to control a smart phone [33]

(d) Non-invasive electroencephalograms sensor system

Using electroencephalograms (EEG) sensor to get electroencephalography is a new technology used in gesture recognition system. EEG sensor can measure brain waves which can be considered in terms of electrical activity. According to the features of brain waves, the corresponding gestures can be recognized. An EEG sensor is shown in Figure 3.10. It is a wearable device with small electrodes to measure EEG signals. With an EEG sensor, the users can perform brain-computer interfacing. Using EEG sensor to recognize gesture involves complex classifying and decoding EEG signals. There are still lots of technical problems need to be solved in this area to improve the practicability and accuracy.

Figure 3.10: A wearable EEG sensor [34]

3.1.2 Vision Based Approaches

With the need for ease and naturalness of user experience, it is necessary to design a system using vision-based gesture recognition not for generic use but under controlled circumstance. As an interdisciplinary area, vision-based approach involves image processing, computer vision and graphics, psychology and machine learning [35]. Different combination of selecting features and recognition algorithms could affect the result of hand gestures recognition [36]. Thus, selecting approaches is significant to recognition. There are different kinds of approaches of vision-based gesture recognition. In terms of the features used to represent the hand, the approaches are divided into two groups: (1) view-based approaches; (2) model-based approaches.

3.1.2.1 View-based approaches

View-based approaches, also called appearance-based approaches, usually model the hand gestures by sequence of views from collection of 2D images and compare specific parameters with image features.

According to different characters used, view based approaches have five types: (a) hand colors and shapes; (b) hand features; (c) SIFT and SURF; (d) mean shift; (e) Viola-Jones algorithm.

(a) Colors and Shapes

Skin color is a vital features to identify and track human hand. However, the drawbacks of this approach is the difficulty of distinguishing other objects that may have the same or similar color with hands or palms, like arm and face. To get high accuracy of color detection, normally the target area would be chosen specially and other parts of body would be covered. Another consideration for color segmentation is the requirement of lighting variations. The structure adaptive self-organizing map (SASOM) neural network, which is used as a new color model, is considered as a powerful representation for efficient image segmentation [37]. With the solution to adjust and transduce color classifiers in

non-stationary color distributions, hand localization can be performed by color tracking with success during experiments.

S. K. Kang et al. presented a gesture detection system to control the non-contact mouse [38]. There are three main steps in the system: (1) color segmentation; (2) hand location; (3) fingertip location.

Figure 3.11: Color segmentation for hand detection [38]

Figure 3.11 shows that without other similar color disturbing, by using color segmentation, hand can be located easily. During the experiment, hand gestures can be recognized with a very high percentage.

For shape-based approaches, there are systems using global shape descriptors to represent various hand shapes. However, one of the drawbacks of global shape descriptors is the computation is too high for real-time system due to the fact that they are pixel-based [39]. The other one is that it requires noise-free image segmentation. J Marnik proposed a method that allows to add any shape into the shapes set, also known as shape descriptors [40]. With this method, environmental effects could decrease by certain extent.

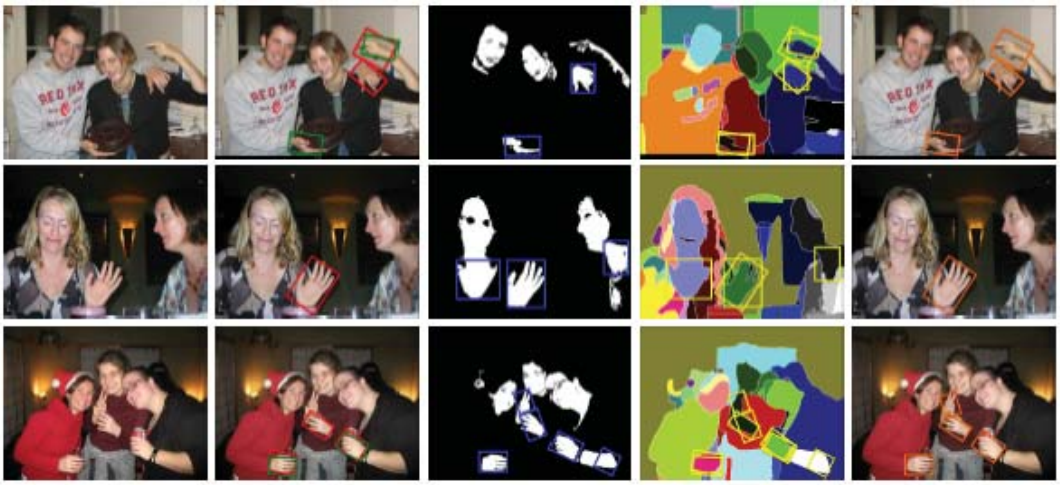

Figure 3.12: Multiple approaches for hand detection [41]

A. Mittal et al. provided a method to detect hands with multiple approaches: context-based, color-based and shape-based detectors [42]. High scoring detections of three datasets has been achieved with multiple approaches, as Figure 3.12 shows, from shape-based approach getting red box in column 2, context detector getting green box, and then color-based approach in column 3 and 4.

(b) Hand Features

Feature-based algorithms involve extracting certain patterns of local pictures and finding the combinations of patterns in the target pictures. Shahzad Malik et al. introduced a tracking system using an efficient corner tracking algorithm [42]. The outline of the system is shown in Figure 3.13. The system is divided into detection mode and tracking mode.

There are six steps in detection mode:

1) Converting the frame into binary image
2) Finding all connected regions which consist of black pixels and meet the requirements for size and patterns.
3) Among the connected regions from last step, finding the four strongest pixels. Making a polygon based on them [42].
4) Computing homography to the created polygon.

5) Generating a picture using unwrapped pixels from the original frame based on homography.

6) Finding the most similar pattern in the new picture.

Tracking mode has three steps:

1) Using search window to computer the current video frame

2) Detecting corner within the search window.

3) Generating a new homography base on the corner locations.

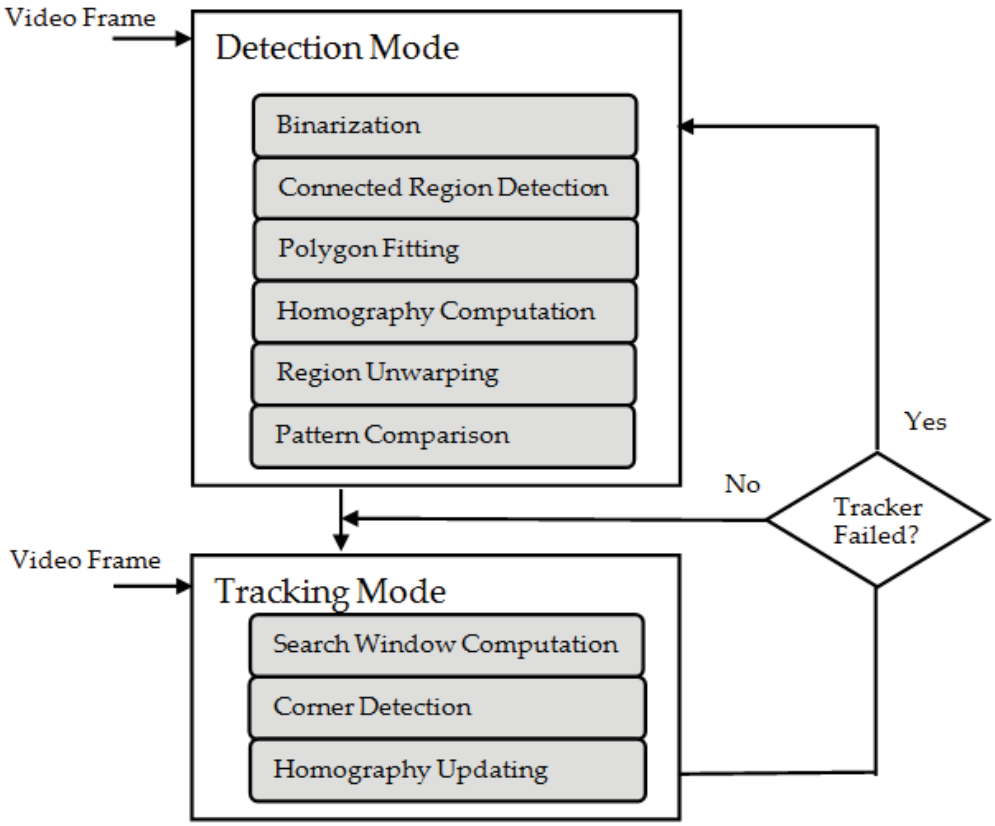

Figure 3.13: The outline of the system [42]

(c) SIFT and SURF

Scale Invariant Feature Transform (SIFT) is a local visual descriptor that is used to acquire invariant features from images for matching with other different views. The algorithm was published by David Lowe in 1999 [43]. Two steps are needed

in the algorithm: detecting feature point and describing feature. However, the requirement of complex computation for SIFT makes it hard to achieve real-time processing [44].

SURF as a speeded up robust features algorithm, uses determinant of the Hessian matrix to detect feature points. Hessian matrix is defined as following:

Where L is the convolution of the Gaussian second order derivation of image at point X(x,y) in scale σ and similarly for L_{xy} and L_{yy}. This algorithm, which was first introduced by Herbert Bay, is partly inspired by SIFT but much faster and more robust than SIFT [45]. Even SURF is fast and strong, the requirement of computation is still huge for real-time processing. To meet the demand for real-time processing, effort has been made to modify Fast-Hessian Detector of SURF by Zhang H et al. [46].

SIFT and SURF are shown to have very similar performance, while SURF is much faster and SIFT outperforms SURF without consideration of speed [47]. As illustrated in Figure 3.14, SIFT gets more matches than SURF in this condition.

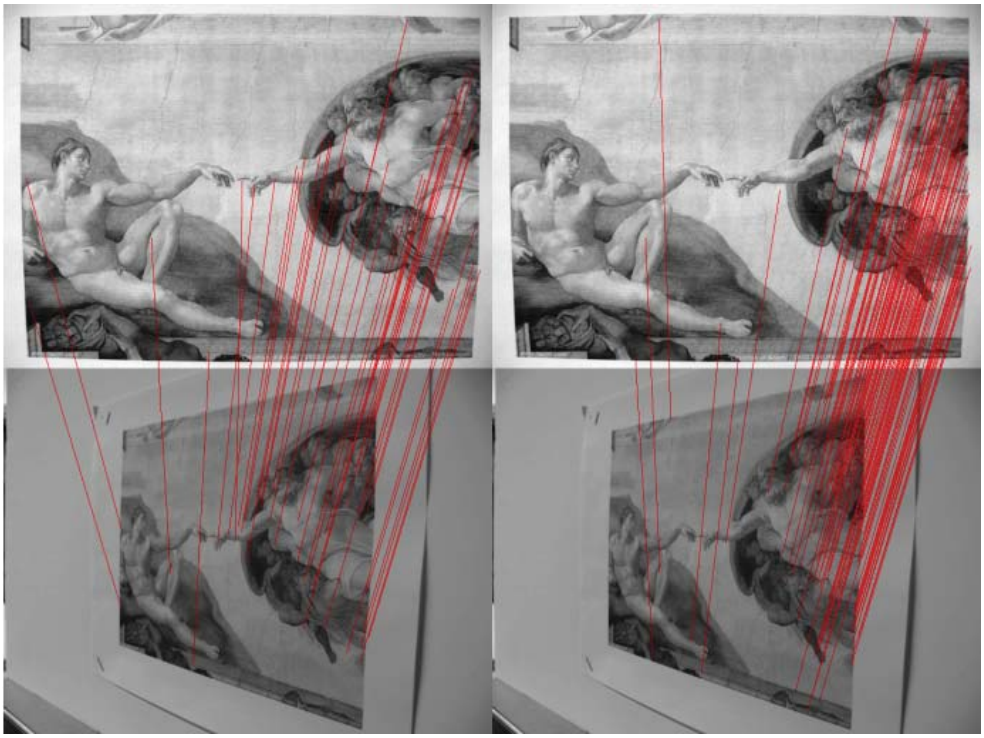

Figure 3.14: SURF vs SIFT [47]

(d) Mean Shift

Mean shift tracking approach is a very successful type in view-based tracking. It's a simple and effective approach, so it is very popular recently [48]. As Figure 3.15shows, the mean shift algorithm is to monitor the shifts from the center of the mass to the center of the region of interest, by using Bhattacharyya coefficient.

However, the drawbacks of mean shift algorithm are also critical: failure in tracking rapid moving objects and recovery ability [49]. In order to deal with the weaknesses, multiple solutions have been attempted for better tracking results. An adaption of the mean shift algorithm, CamShift, the Continuously Adaptive Mean Shift Algorithm, was used by Chetan. S et al. for gesture recognition system, which could track on type of feature spaces [50]. With the combination of particle filtering, the mean shift embedded particle filter (MSEPF) was used for improving the sampling efficiency and tracking rapid motion [52]. As illustrated in Figure 3.16, top row is the result of using MSEPF, middle one is using particle filtering approach and the bottom row is using mean shift only. By comparison with other two, MSEPF shows more accurate than single approach.

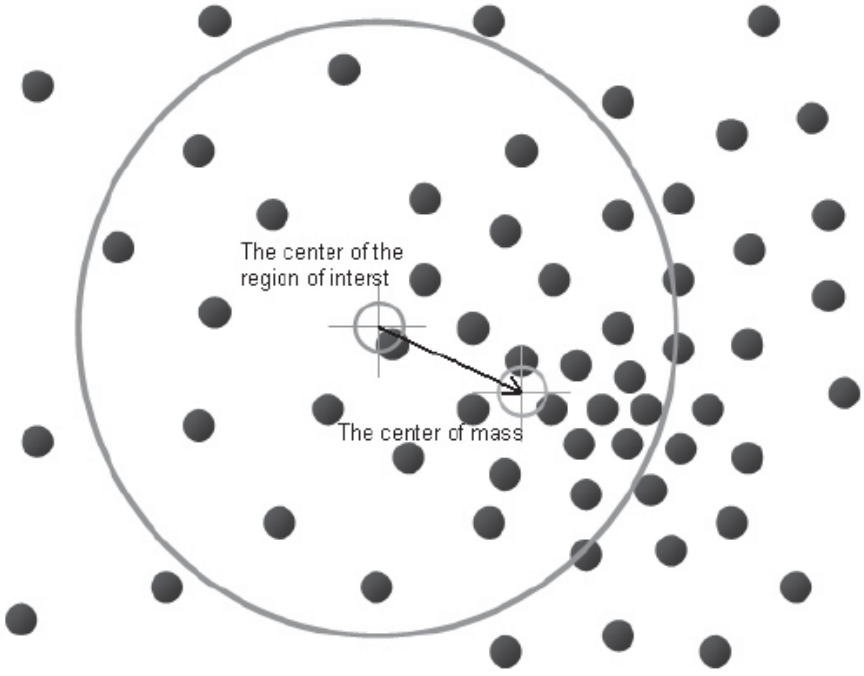

Figure 3.15: The mean shift algorithm [51]

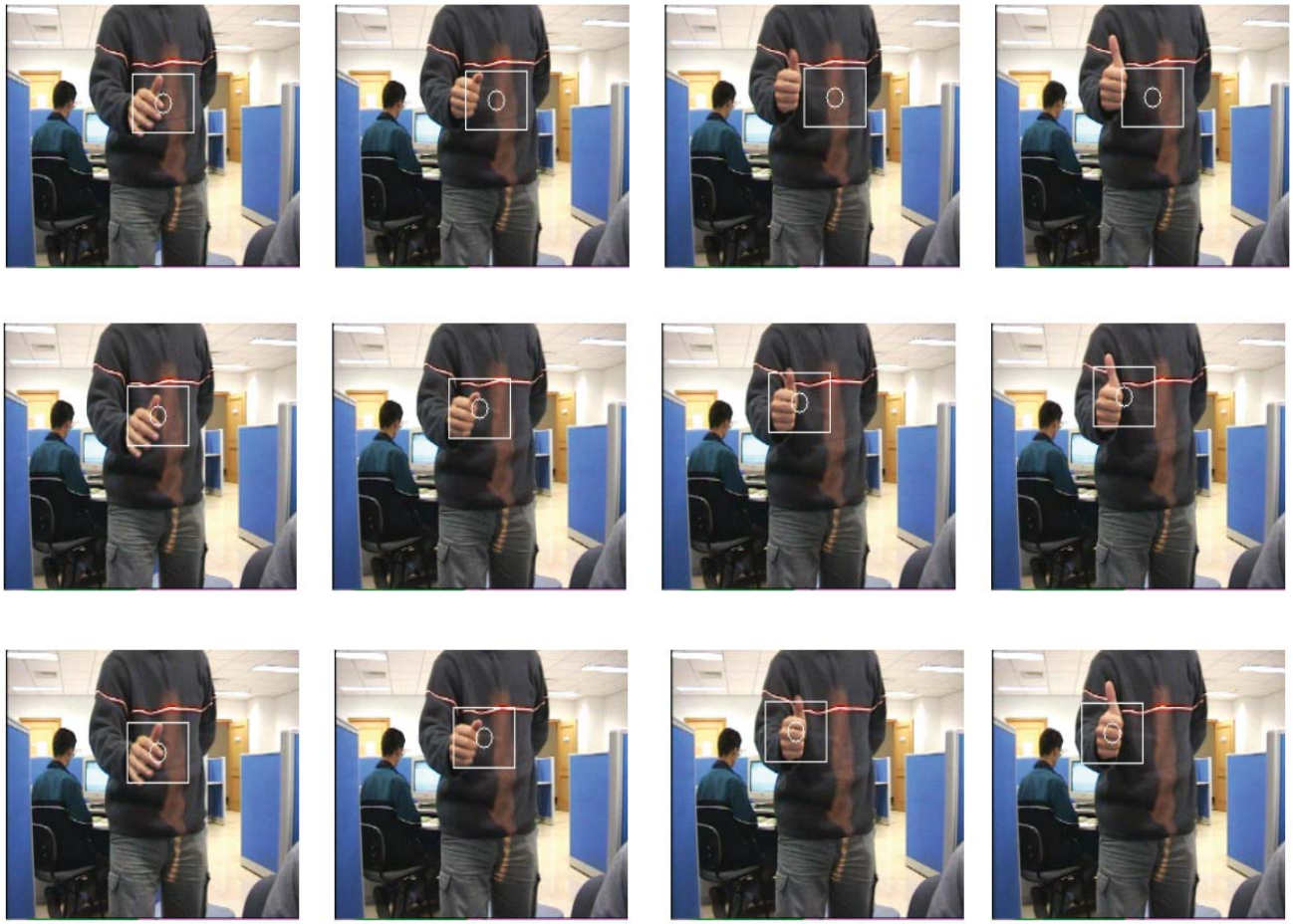

Figure 3.16: Sampled tracking results [52]

(e) The Viola-Jones Algorithm

The Viola-Jones Algorithm is the first object detection framework with relative accuracy for real-time operation proposed by Paul Viola and Michael Jones in 2001 [53]. In the algorithm, "Integral Image" is introduced to compute rich features, and a learning algorithm based on AdaBoost (Adaptive Boost) is used for feature selection. The Viola-Jones algorithm is motivated for face detection and has been implemented in OpenCV as cvHaarDetectObjects(). For real-time gesture detection and tracking, Viola-Jones algorithm is also a good choice, as Qing Chen et al. proposed [54]. With two-level approach, different hand postures can be recognized as Figure 3.17 shows.

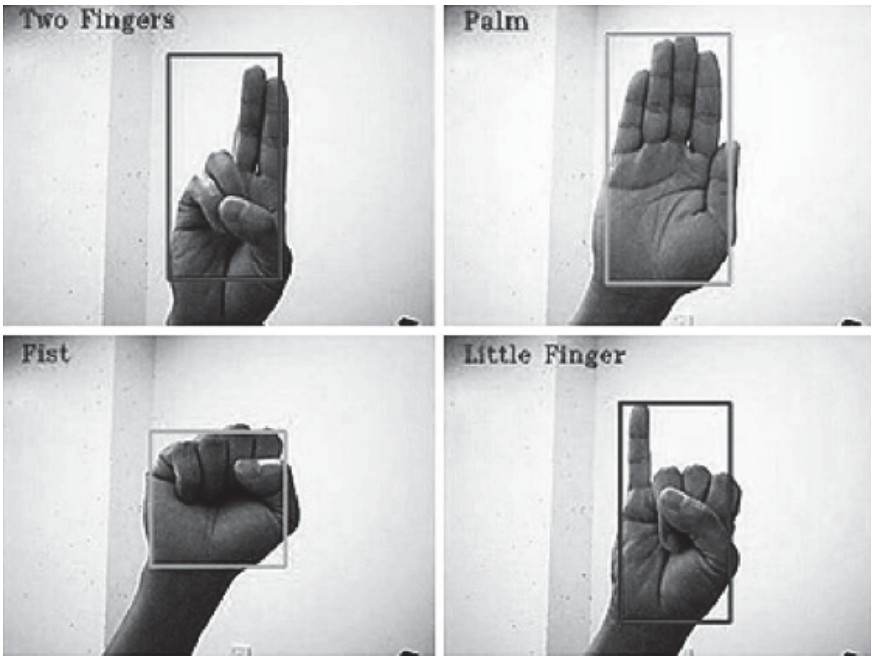

Figure 3.17: Recognition results for hand postures [54]

3.1.2.2 3D Model-based approaches

In contrast with 2D appearance-base approaches, 3D approaches use complex models to represent object hand. Recovering the parameters of object hand involves an estimation-by-synthesis strategy. This strategy suggests to align the features of 3D model with the input images which are captured by the camera, and minimize the variance between them. This is a very difficult technical problem as human gesture is very complex. Basically, pictures which are captured in different position and capable to present the key features are essential to modelling process. Figure 3.18 shows a general process of 3D hand modelling.

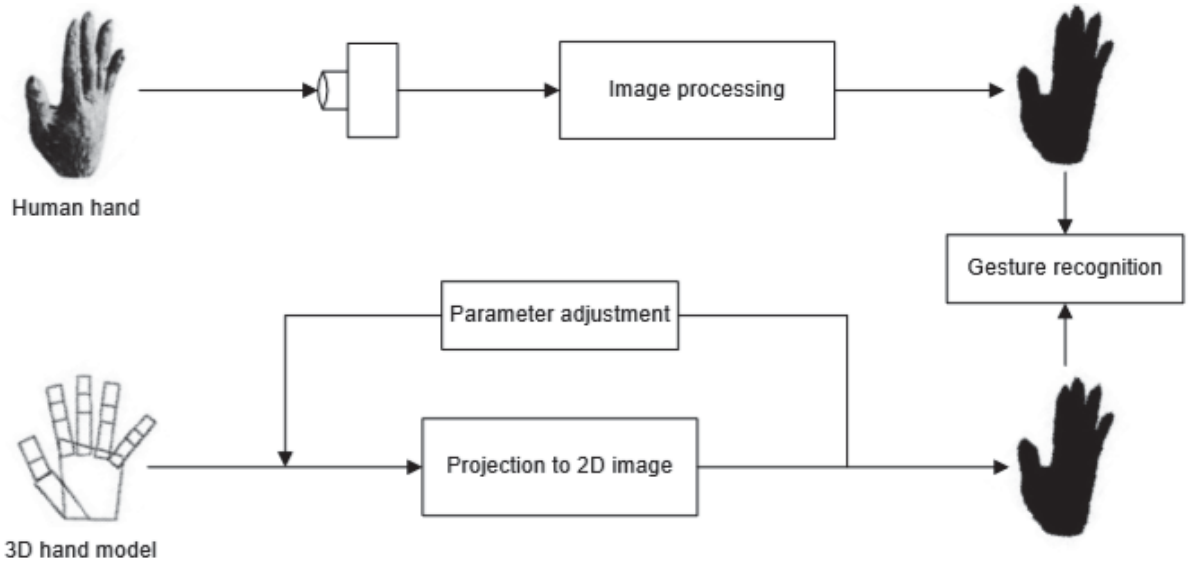

Figure 3.18: Process of 3D hand modelling [55]

(a) Analysis-by-synthesis

Analysis-by-synthesis is used in many gesture recognition systems. Take the project of Carlo Tomasi et al. as an example, they developed a 3D tracker which is able to track and model hand in fast and complicated motions [55]. They used 2D-based method to classify images and 3D-based method to do motion interpolation. They have developed the prototype which can satisfy basic requirements under certain conditions. But when the lighting conditions changes, the accuracy will vary and become more appropriate to track. Figure 3.19 shows the result of hand tracking system when the hand is moving and changing gesture on the same time. The system analyzes the gesture from the frame and find the most-like 3D-model in the database. The left of Figure 3.19 are several real-time video frames, and the right are the corresponding 3D-models in the database.

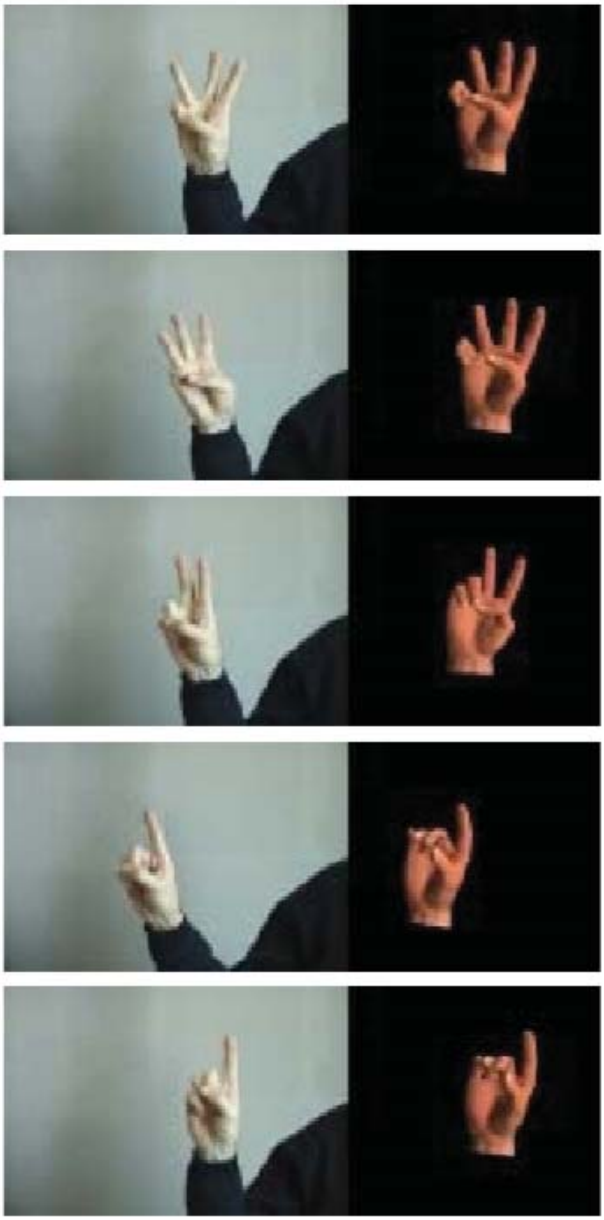

Figure 3.19: Example of 3D hand tracking system [55]

(b) Image retrieval

Image retrieval represents the method to do 3D hand model based on the comparison of input image and database. The database contains reasonable amount of relevant images and system can find the best match in the process.

H. Zhou et al. [56] proposed a recognition system based on Okapi-Chamfer matching algorithm. They used inverted index method to develop the database and implement the matching algorithm to accomplish highly efficient object recognition. Figure 3.20 shows the retrieval results of some target images. The first column from the left shows the query images which are real-time frames. The left columns are corresponding retrieval results. The size of the retrieval image is defined by the similarity and the biggest image is the most similar.

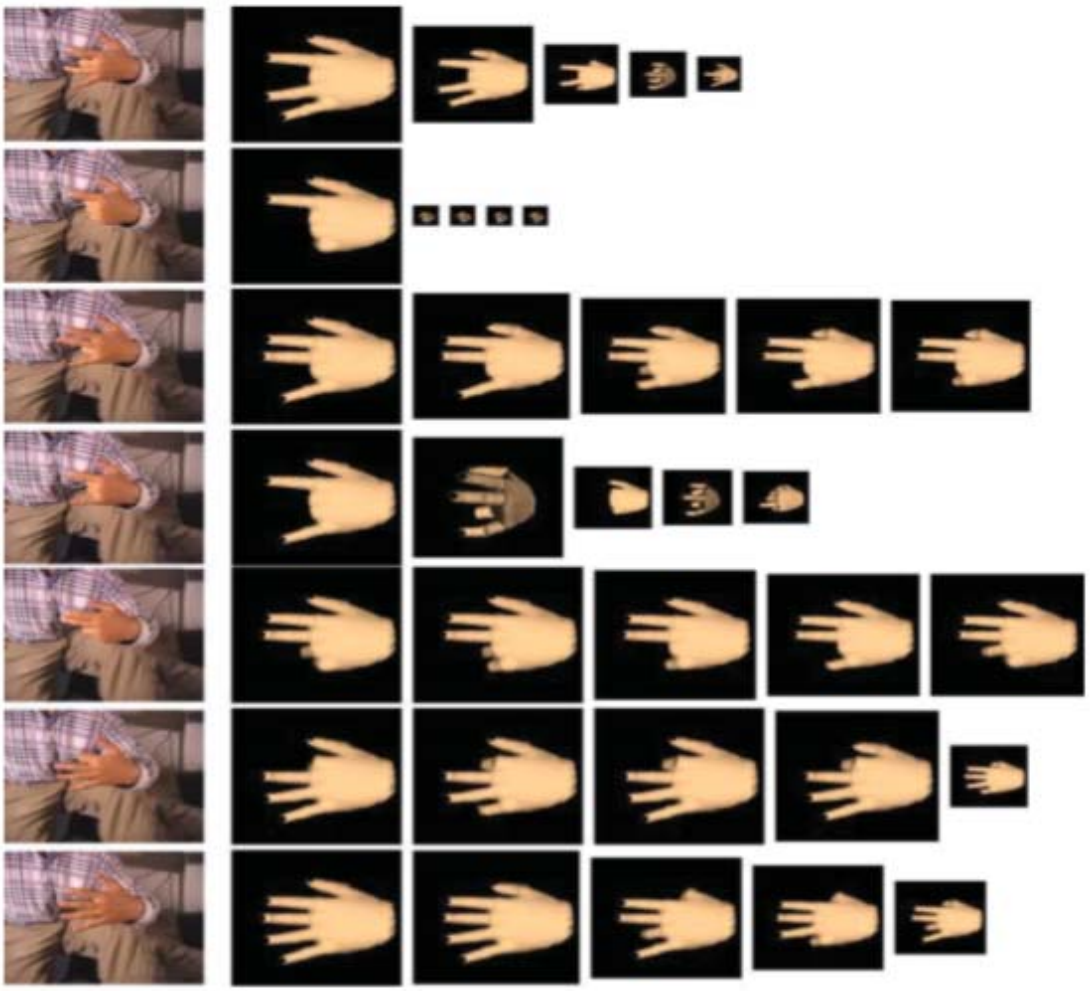

Figure 3.20: Retrieval results of sample images [56]

3.1.2.3 View-based versus 3D model-based

Basically, view-based approaches are easier to be implemented in applications while 3D model-based approaches involve complex signal processing which requires strong ability of computation and database support. But 3D model-based approaches can achieve high accuracy and are able to applied in various conditions.

View-based approaches require relatively less processing time and computation cost, so they are practical to perform real-time recognition. But compared with 3D model-based approaches, view-based approaches require strict conditions of lighting and angles.

3.2 Voice Control

With the technologies getting more and more human-centered, voice control is becoming very mainstream nowadays. There are many applications using voice recognition to control smart devices. In this section, we will review voice recognition from three aspects: classification of voice recognition, the process of voice recognition and three mainstream algorithms.

3.2.1 Classification of Voice Recognition

Voice recognition systems can be classified into three types: (1) isolated speech; (2) discontinuous speech; (3) continuous speech.

3.2.1.1 Isolated speech recognition

Isolated speech recognition is to recognize single word of users and it is a relatively easy approach. Each spoken word has its information in form of speech signals. Figure 3.21 shows an example of speech signal in MatLab. It is the word 'No' presented in time domain.

There are both useful sounds and "noise" of speech signals. To separate these two parts of human speech, using the cepstrum is an efficient method [57]. MFC (Mel frequency cepstum) is a classic method to transform speech signal into cepstral. The coefficients of MFC are MFCC (mel-frequency coefficients) [57]. S. D.

Dhingra et al. [58] used MFCC (Mel frequency cepstrum coefficients) and DTW (Dynamic Time Warping) to recognize isolated speech. DTW algorithm is an approach to calculate the similarity between two time series which may vary in time or speed [60].

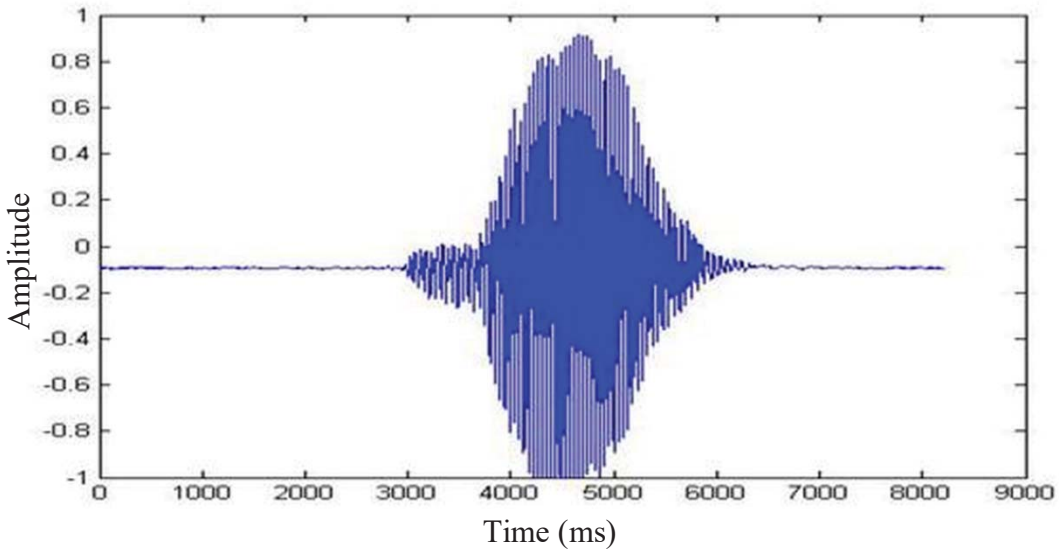

Figure 3.21: An example of word 'No' in Matlab [59]

3.2.1.2 Discontinuous speech

Discontinuous speech is similar with the isolated speech. There are intentional pauses among sentences and may be more than one person in the conversation. Annalisa Sannino et al. [61] proposed a new methodology to analyze discontinuous speech in European Union conversations. This methodology identifies topical changes according to the interventions among sentence and also when the speaker changes. And according to the topical changes, the incessant discursive fluctuations can be distinguished [61]. This is the main difference compared to other methods that all elements are translated.

3.2.1.3 Continuous speech

Continuous speech represents the natural performance of people's speech. There are no pause on purpose. It is the most difficult situation to recognize speech. The technology in this area has not developed well yet and the current methods are obviously short of accuracy.

3.2.2 Process of Speech Recognition

A typical process of speech recognition is shown in Figure 3.22. There are six steps in the process.

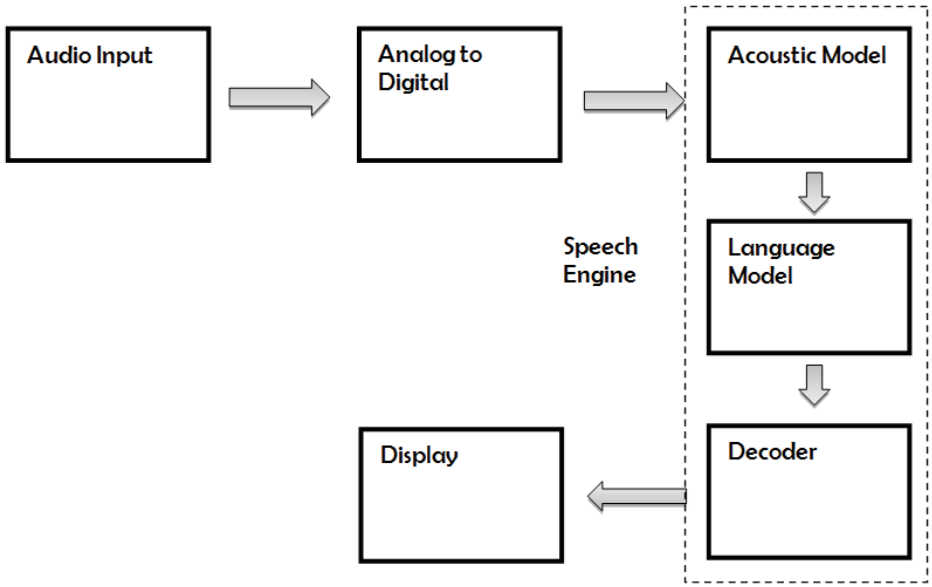

Figure 3.22: Process of speech recognition [62]

3.2.2.1 Audio input

The first step is to receive audio signals through audio input devices [62]. Microphone is the most common audio input device which allows users to speak to the speech recognition software.

3.2.2.2 Digitization

In this stage, the analog signals collected are converted into digital form. There are two steps: sampling and quantization [62]. Sampling is the process to transfer continuous signal into discrete signals while quantization is the process of mapping a big set of signal values into a small set.

3.2.2.3 Acoustic model

The main task of acoustic modelling is to establish a file which contains statistical representation for the speech. Hidden Markov Model is a widely used method in this area. It is the essential requirement of recognition using speech engine.

3.2.2.4 Language model

Language Modelling is used to capture the properties of a certain kind of language. Then based on that the next word in a sequence can be predicted which can help to distinguish between speeches that sound similar [62].

3.2.2.5 Decoder

A typical speech engine consists of acoustic modelling, language modelling and decoder. After the input audio is converted in proper format, the next step is to decode it into corresponding text [62].

3.2.2.6 Display

In many applications, the recognized speech are displayed at last. But this is not the essential step.

3.2.3 Algorithm

There are basically three kinds of most popular algorithms: (1) Dynamic Time Warping (DTW); (2) Hidden Markov Model (HMM); (3) neural networks.

3.2.3.1 Dynamic time warping

Dynamic time warping (DTW) is a historical algorithm than HMM. It can measure the similarity between two time series which may vary in time or speed [60]. For example, to detect the similarities of two walkers. The similar patterns can be detected no matter of the change of speed during the process. Although nowadays many developers prefer HMM method than DTW, there are still various applications based on DTW techniques. Thite et al. [63] compared DTW, neural network and HMM in the experiment which was to recognize isolated speech. The recognition accuracy achieved 95% when using DTW and it was the highest among the three algorithms. The accuracy of three algorithm is shown in Table 3.1.

Table 3.1: The accuracy of three algorithms [63]

Algorithm	Vector size	Recognition accuracy
DTW	35	95%
Neural Network	35	84%
HMM	35	93%

3.2.3.2 Hidden Markov Model

Hidden Markov Model is a powerful statistical tool which is used in many applications for signal recognition, especially in speech recognition. HMM is based on the mathematics which was developed by L. E. Baum and coworkers [64]. HMM based system often involves Viterbi algorithm which is a dynamic programming algorithm to find the most similarity in sequence of hidden states . But there are some exception like N. Najkar et al. 's project. They introduced a new decoding method based on HMM. A search method is applied to replace dynamic programming which is the common way to determine the most acoustic sequence in the input speech signal [65]. The search method uses particle swarm optimization which is a population based evolutionary algorithm. The main concept is to create an initial population of segmentation vectors in the solution search space to improve and correct the location of segments [65].

3.2.3.3 Neural networks

Unlike HMM, neural networks can model complex non-linear relationships. This method can calculate the probabilities of speech segmentation by discriminative training [66]. It is much more effective in short-time unit recognition than continuous recognition. So there are many applications which used neural networks to classify phoneme and recognize isolated speech but rarely successful cases to recognize continuous signals.

In recent years, deep neural network (DNN), which contains multiple hidden layers to provide better learning capacity, has been widely used in deep learning speech recognition. With the success of DNN, deep learning methods are now blooming in both academic and industry across. Deep learning methods can be found in almost all commercial speech recognition systems.

3.3 Related Work

3.3.1 Natural User Interface Using Color markers

J. J. Choondal and C. Sharavanabhavan [67] introduced a new system of natural user interface. This system can recognize gestures and keep tracking them by locating the selected fingers. They use four different color markers which are mounted on four fingers.

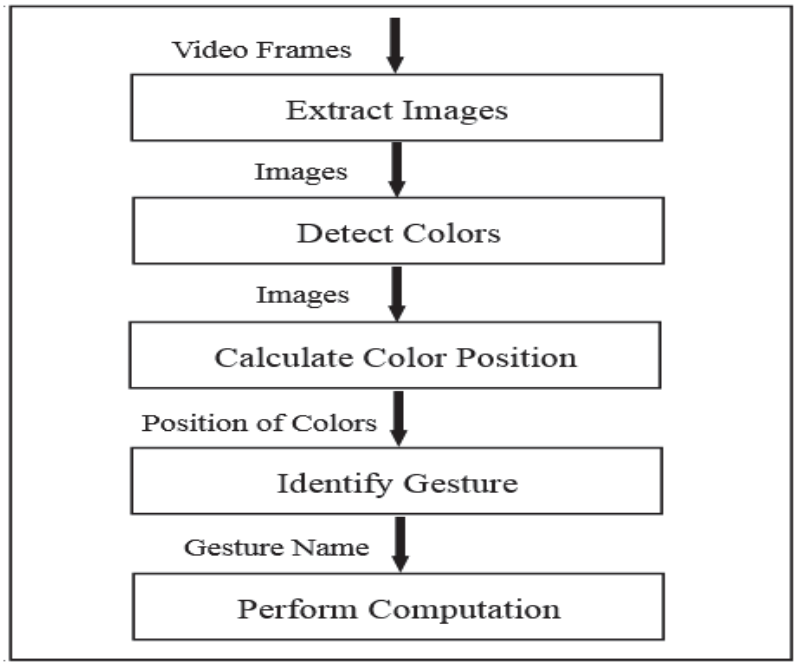

Figure 3.23: Natural user interface system [67]

The process of recognition basically consists of five steps as shown in Figure 3.23. First, extracting images from the input real-time video using a simple web camera. The input video is transformed into frame sequences. Then each frame is converted into proper format as pre-processed images. Then, system detects color markers in pre-processed images. Third, when the result of detecting color is positive, the system calculates the positions of each color markers and the relative positions. Then, based on the position data, the system can identify gestures. The corresponding execution is then performed.

In this paper, the color markers are blue, yellow, green and red. After the input images are transformed into HSV (Hue-Saturation-Value color model) space, the system can identify and locate the color markers. They designed a method to identify the positions of color markers based on evolutionary process model.

Evolutionary models inherits the concept of evolution and are iterative into software development. This kind of model enhances the ability to develop much more complicated versions of the software. Evolutionary process model regards the development as serial heaps and each standing for an independent loop of the spiral model [67]. The advantage of this model is that the process will not restart from the beginning when failures occur. The action is re-executed from the place failure occurred and this obviously improves the efficiency of the process.

Figure 3.24 shows the gesture for zoom operation. Adding or changing the gesture commands in this system are very simple and this makes it very practical for variable situations.

The approach introduced in this thesis is easy and effective. The recognition can reach high accuracy under suitable environments. The conditions for suitable environments are quite strict: the lighting conditions which might affect the accuracy greatly; the background should not have the same color of the markers.

Figure 3.24: The gesture used to zoom [67]

3.3.2 Cursor Control Using Haar Classifier

K. Madhuri and L. P. Kumar developed a real-time hand gesture recognition system to control cursor which can recognize gestures through web camera and execute corresponding cursor functions [68].

The algorithm used in their project is Haar Classifier [53] which is a widely used algorithm in this area and is devised by Viola and Jones. For this project, developers trained their own classifier first. There are two sets of images used to train: positive images and negative images. Positive images are proper images which contain the correct features and negative are images without the target features. In this case, the target feature is palm. In order to get perfect classifier, great amount of various images are needed and this train process may take long time. Some samples of negative images (images have no palm) are shown in Figure 3.25 and some positive images (images have different palms) are shown in Figure 3.26.

Figure 3.25: Samples of negative images [68]

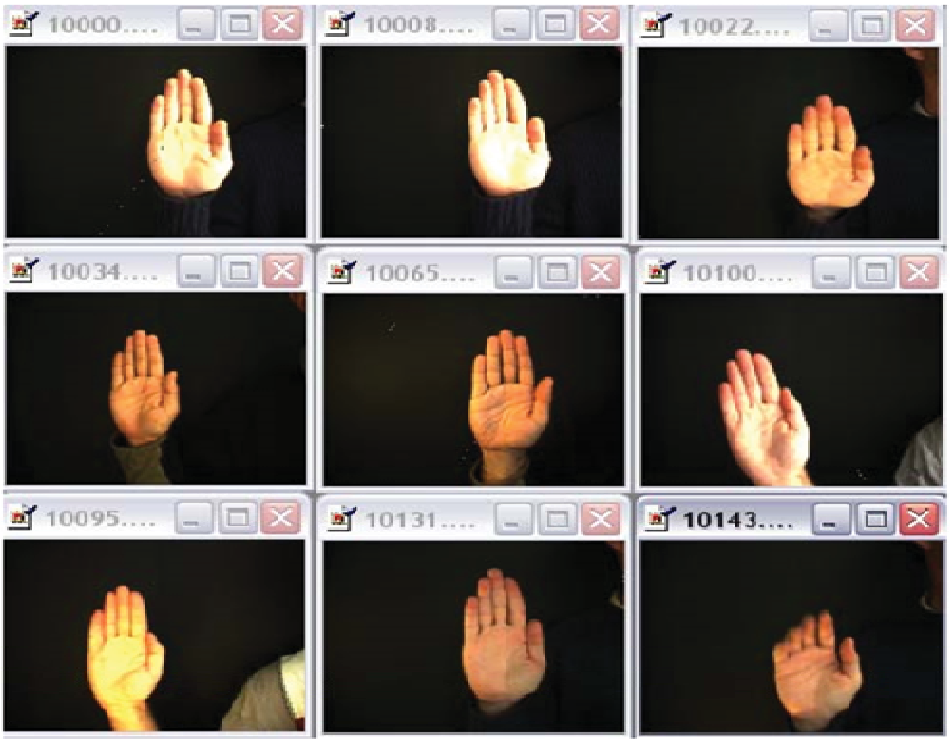

Figure 3.26: Samples of positive images [68]

The process is divided into three stages. First is pre-processing which involves denoiser and converting into HSV space. Denoiser is to filter the original images to achieve higher accuracy in the following process. Converting the captured images into HSV space ensures normal color recognition under different kinds of lighting conditions. The second stage is hand recognition which is to recognize the palms and get the coordinates of them. Then according to the movements of gesture, the cursor can execute the proportionate movements. The flow chart of the process is shown in Figure 3.27.

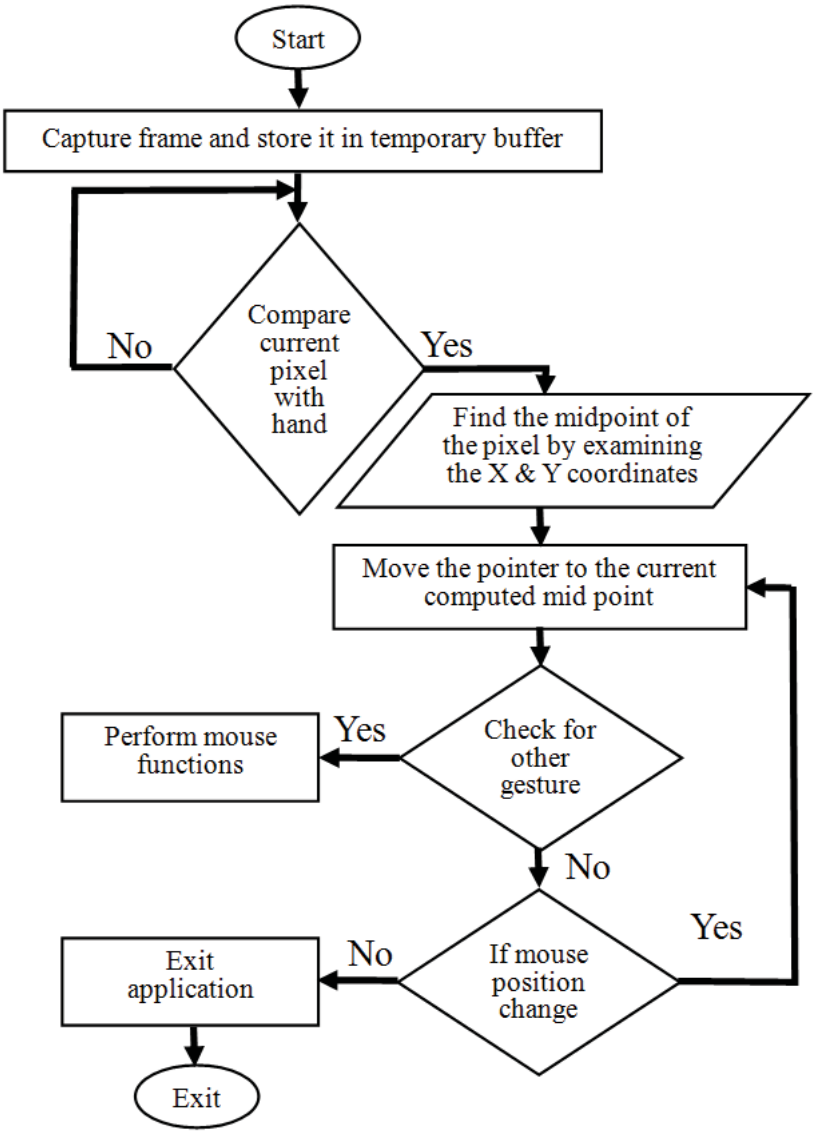

Figure 3.27: Flow chart of process [68]

The approach introduced in this project is more flexible and able to reach higher accuracy under different environments compared to J. J. Choondal and C. Sharavanabhavan's. But the efficiency is lower due to the complexity of the algorithm. The system needs much more time to recognize gesture and train for new added gestures.

3.3.3 The Architecture Using Both Statistical and Syntactic Analysis

Q. Chen [51] proposed a new architecture to perform real-time gesture recognition and hand motion tracking. The architecture can be decoupled into two levels which combines both statistical and syntactic analysis.

Figure3.28 shows the architecture. The task for low-level is to detect and track gestures using Haar classifier with high efficiency. Web camera is used to capture real-time images and the resolution is 320×240. Next step is to pre-process the input image including segmentation, denoising and transferring to proper format. Then feature extraction is done to clean the image for classifiers. The classifiers are trained using two sets of positive and negative images. And classifiers can detect, recognize and keep tracking gestures.

In high-level, the gesture detected from the first level is compared with the defined patterns using SCFG (stochastic context-free grammar) which can greatly improve the accuracy [51]. This stage includes two parts: local finger motion and global hand motion analysis. They both use defined grammar which represents the relationship between the gesture captured and the corresponding one in database.

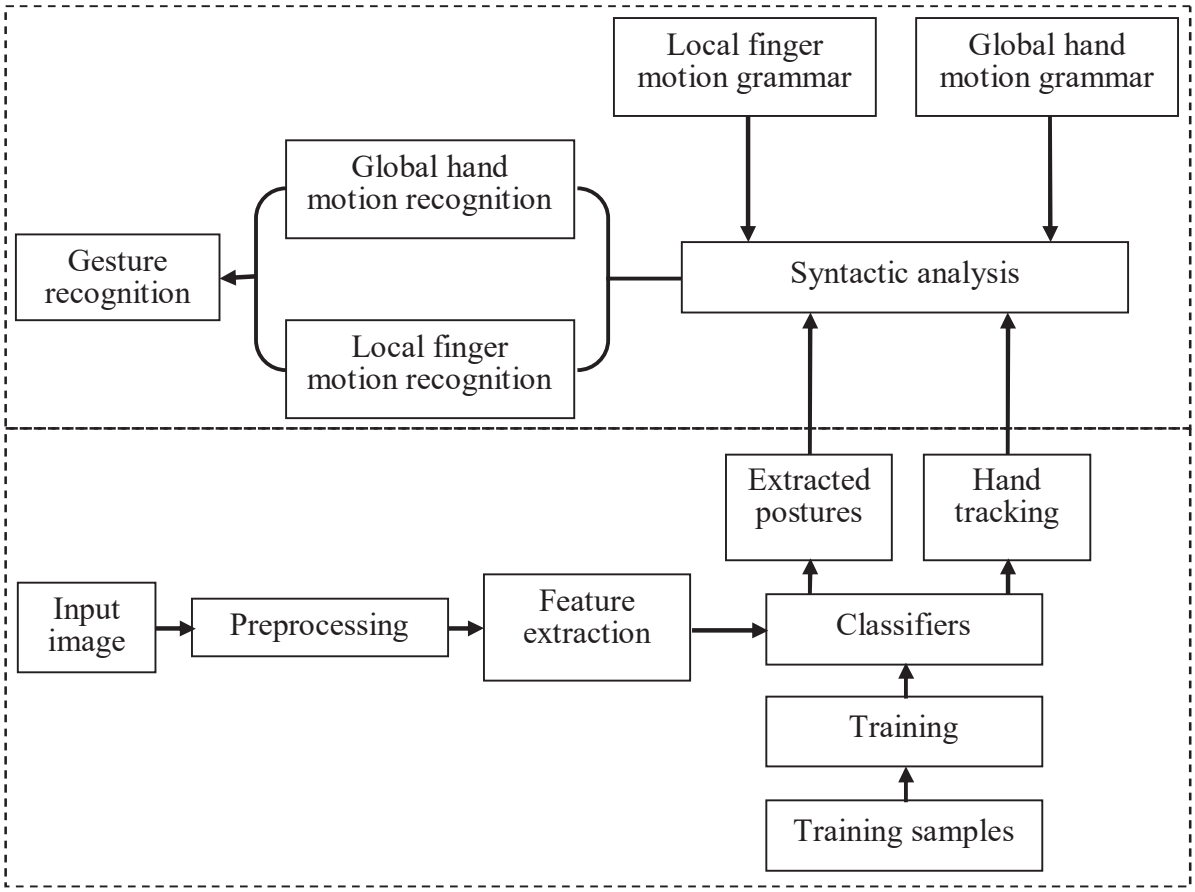

Figure 3.28: Architecture of system [51]

This system uses both 2D and 3D methods to track hand and recognize gestures. It can reach high accuracy like 3D gesture recognition while remain efficient like 2D gesture recognition.

3.3.4 Hand Motion Recognition Using Kinect

K. Rimkus et al. [69] introduced a Hand Motion Recognition System using a Kinect device which can capture depth images for calculating 3D coordinates. In this paper, they used ten different single hand gestures for experimentation.

The Kinect sensor is shown in Figure 3.29. It has a laser-based infrared (IR) pattern projector, an IR camera and a RGB camera. It can triangulate points in space like a depth map scanner. The Kinect can deliver three outputs: IR image, RGB image, and (inverse) Depth image. And it can be connected directly via the USB interface.

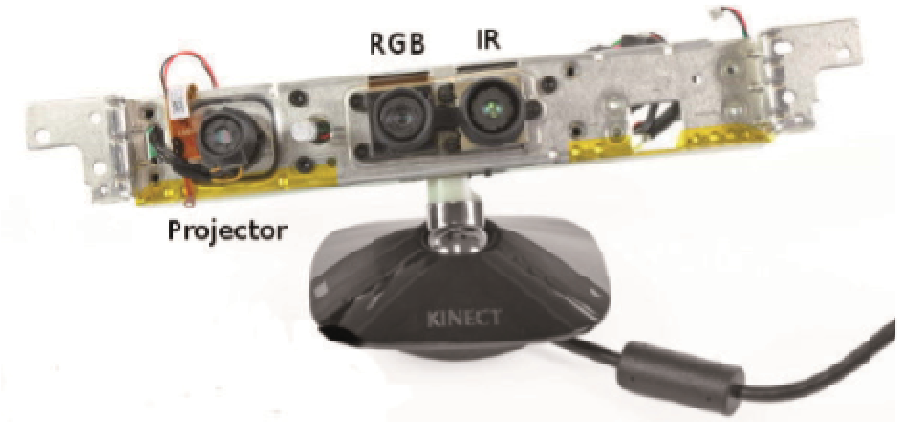

Figure 3.29: Kinect device [69]

They designed ten gestures as commands for robot. Each gesture is trained based on the data gathering from seven people. The defined capture duration for gesture equals to the time of 15-16 frames, and there are about 0.1s intervals between the frames. Figure 3.30shows the samples gathered and structured in 3D space.

As the distance between users and Kinect affects the coordinates data, the differences of coordinates are used instead of the coordinate value. The typical formula is [69]:

$$k_{xyz}=[c_{xyz}(2)-c_{xyz}(1),\ c_{xyz}(3)-c_{xyz}(2),\cdots,\ c_{xyz}(n)-c_{xyz}(n-1)]$$

where k_{xyz} is the gesture vector and $c_{xyz}(n)$ are gesture coordinates at time moment n in 3D space.

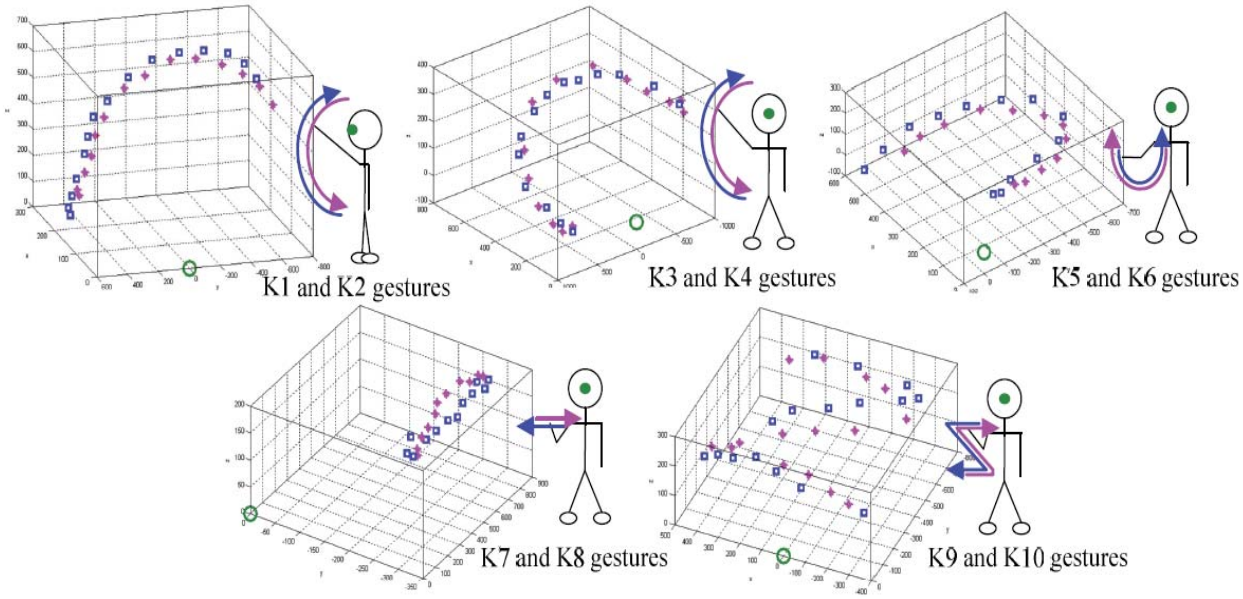

Figure 3.30: Gestures and their representation in 3D [69]

In contrast with J. J. Choondal and C. Sharavanabhavan's method which is based on 2D recognition, this 3D approach is much more accurate and practical under different environments. But the flexibility is lower when adjusting gesture commands.

3.3.5 Gesture Control of Smart Home

R. Neßelrath et al. [70] proposed an interaction for smart home based on gesture recognition which focuses on using simple gesture to control multiple smart appliances.

All appliances can be accessed by a middleware which is based on the ISO standard URC (Universal Remote Console). Using URC, developers can develop different user interfaces vary with different environments.

This project is not vision-based and uses the WiiMote as input device. The WiiMote is a controller which contains three accelerometers. Accelerometers can detect the movements of user which can be used as gesture patterns for controlling.

In this paper, they proposed a concept that one single gesture is used to trigger multiple appliances which can make gesture control more efficient and flexible. Table 3.2 shows seven predefined categories used in smart home. Each one has its unique gesture.

Table 3.2: Predefined categories [70]

Gesture	Functions	Impact
Switch on	Switch on television Switch on standard light Switch on fan Switch on light of the hood	Appliance turned on
Switch off	Switch off television Switch off standard light Switch off fan Switch off light of the hood	Appliance turned off
Increase	Decrease television volume Decrease fan speed Decrease light setting	Appliance setting decreased
Decrease	Decrease television volume Decrease fan speed Decrease light setting	Appliance setting decreased
Next choice	Television channel	Switch to next channel
Previous choice	Television channel	Switch to previous channel
Toggle mute	Television	Toggle between mute/unmute

The system is non-vision based which can achieve higher efficiency than vision-based system. It is ideal for elder people who prefer to simple gesture commands. But for young people, non-vision based system is not human-centered enough and less flexible to control multiple smart devices.

3.3.6 Summary of Related Work

J. J. Choondal and C. Sharavanabhavan [67] introduced an easy and effective approach to realize gesture control. The process of recognition is relatively quick and accurate under certain conditions. The basic principal of recognition in this case is based on color recognition. The biggest disadvantage of this approach is that it has high requirements on lighting conditions. When the lighting condition changes, the accuracy will great decrease. Background is the other big restriction.

When the same color of markers shows in the background, the system may get confused and take it as the target object.

In contrast with skin-based algorithm, K. Madhuri and L. P. Kumar's project [68] is more flexible in different conditions using Haar classify. It can capture and keep tracking gestures in high accuracy. One of the drawbacks in this project is the requirement of distance between users and camera. As the system must capture the clear shape of the hand, it cannot remain effective in long distance. The rate of recognition is also need to be improved.

Q. Chen [51] introduced a hand tracking and gesture recognition system which is both accurate like 3D gesture recognition and efficient like 2D gesture recognition. The architecture of his approach is two-level and combines statistical and syntactic analysis.

By comparison with 2D gesture recognition, K. Rimkus [69] uses 3D algorithm in his system which is very accurate and practical under various backgrounds. But 3D algorithm needs to analyze abundant data which leads to much more processing time. One shortcoming of this algorithm is the less flexibility when adding new gestures. There are lots of preparation needed including collecting images from different persons in different angles, training data and analyzing data.

R. Neßelrath et al. [70] proposed a system which can realize gesture control of smart appliances using simple and few gestures. According to the test, the response time is reasonably shorter than vision-based system. Developers create rule which can control several appliances with only one gesture. This makes the system more humanized and applied to elder people.

CHAPTER 4 SYSTEM DESIGN AND IMPLEMENTATION

In order to employ the gesture and voice control into the Internet of things, a prototype is constructed as shown in Figure 4.1. This prototype contains a local network system which is centered on an embedded system based on a development board. A camera and a voice recognition module are used as input devices for the embedded system. With the gesture and voice as input methods, the development board processes the control commands and transfers them into the actuators via wireless connection. The wireless connection technology used in the prototype is ZigBee. Two ZigBee modules are used in the system as a transmitter and a receiver respectively. Devices are connected with relays and controlled by the signal from the ZigBee receiver module.

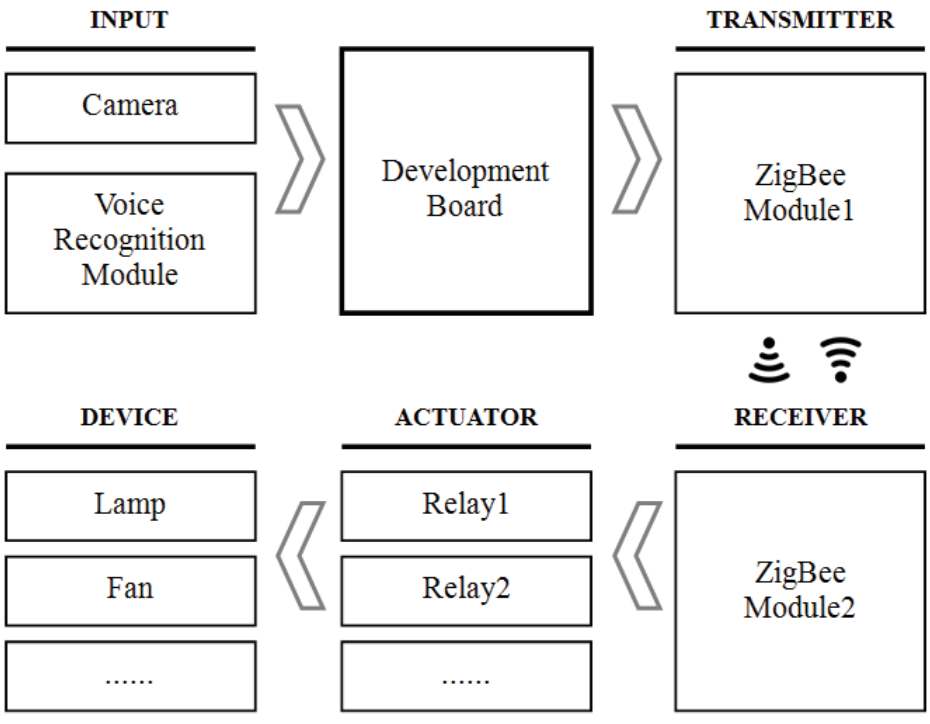

Figure 4.1: Overview of the system design

As the development board integrates a RJ45 (10M/100M) Ethernet interface, the prototype can be connected to the Internet through a gateway as shown in Figure 4.2. There will be huge potential for further study in cross-network or Internet network control. The data collected from the devices can be sent to the cloud. The mobiles and computers can control the devices by sending commands to the development board. In this thesis, only local network system is constructed and discussed.

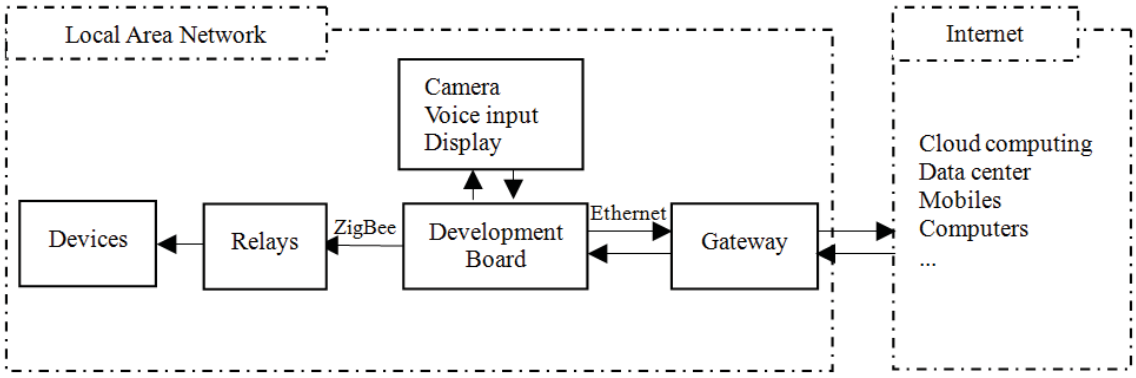

Figure 4.2: Local area network connected to the Internet

To explain the whole system, the following parts are introduced in this chapter:

(1) Algorithms for gesture control and voice control. Three algorithms for gesture control and the algorithm for voice control are presented.

(2) The embedded system platform. The profile of embedded system, development board system and the cross compilation environment we used are introduced.

(3) Hardware design. The hardware used in the thesis are introduced: ZigBee module, USB camera, voice recognition module, relay and target devices.

(4) Software design. The process of building the software system is introduced and some main functions are discussed.

4.1 Gesture Control and Voice Control Algorithms

4.1.1 Gesture Control Algorithms

Three main algorithms for gesture control in the thesis are introduced :

1. Border following algorithm. This algorithm is used to get the contours of the user hand. By analyzing the contours, we can get the information to recognize gesture.

2. Convex hull algorithm. We use this algorithm to extract the convex hull of the given contours. Convex hull is easier and more suitable to analyze in this case.

3. Ramer-Douglas-Peucker Algorithm. This algorithm is used to simply the convex hull we get.

4.1.1.1 Border Following Algorithm

Also known as contour tracing, it is an algorithm to extract the boundaries from a given image. The contour represents the features of the pattern. Using contour analysis is a common method to recognize gesture. In this project, we use S. Suzuki's algorithm [71] . It can extract the outermost border in a binary image.

Figure 4.3 shows the contours of some given images. The contours of different objects are indicated with red or blue curves.

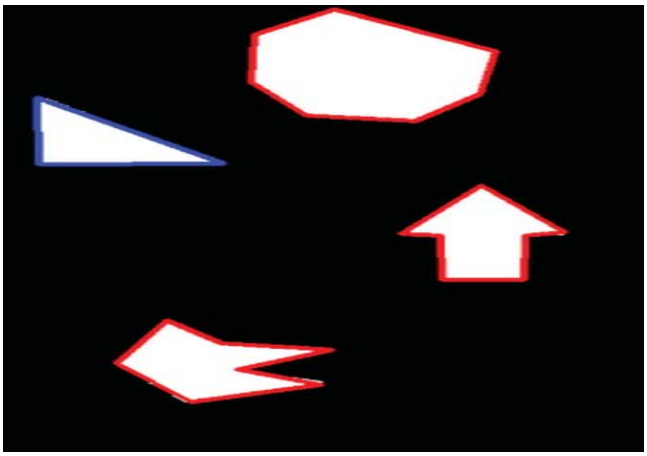

Figure 4.3: An example of contours of images

■ Basic Concept

(a) Digital image

A digital image can be described in a 2D space using a[m,n] which represents N rows and M columns [72]. Every intersection denotes a pixel which in binary image can only be 0 or 1. Pixels in binary image are called 0-pixel and 1-pixel, and generally, it is assumed that 0-pixels are filling the frame. The pixel assigned in certain row and column is represented using its coordinates [i,j]. The density of a pixel is represented by $F=\{f_{ij}\}$ in which represents the row number, j represents the column number and f_{ij} is the density value. The border in a binary digital image is between the connected components of 1-pixels and 0-pixels. Normally 0-pixels are regarded as 8-connectivity (4-connectivity) [75] while 1-pixels are regarded as 4-connectivity (8-connectivity) [71].

(b) Definitions for border and surroundness

1. Border point: if there is a 0-pixel in a 1-pixel's 8-connectivity (4-connectivity) neighborhood, this 1-pixel is defined as a border point.

2. Surroundness among components: for a binary image, if S_2, which is one of its two connected components, has a pixel belongs to the frame, S_2 is said to surround S_1. In the mean time, if there is a border point between S_1 and S_2, we can say S_1 surrounds S_2 directly [71].

3. Outer border and hole border: for given a binary image with 0-component S2 surrounds 1-component S1 directly, the set of the border points is called the outer border. Otherwise, if a 1-component S1 surrounds 0-component S2 directly, it is called the hole border. The border points that compose the outer border or hole border are all 1-pixels.

4. Parent border: as shown in Figure 4.4, for given a 1-component S_2 and S_5 surrounds 0-component S_1 directly, S_1 is the background while B_1 and $B_4$are outer border. As S_2 is a hole, B_2 is the hole border. 1-component S_4 surrounds

hole S_3 directly, so B_3 is the outer border and defined as a parent border of hole border B_2.

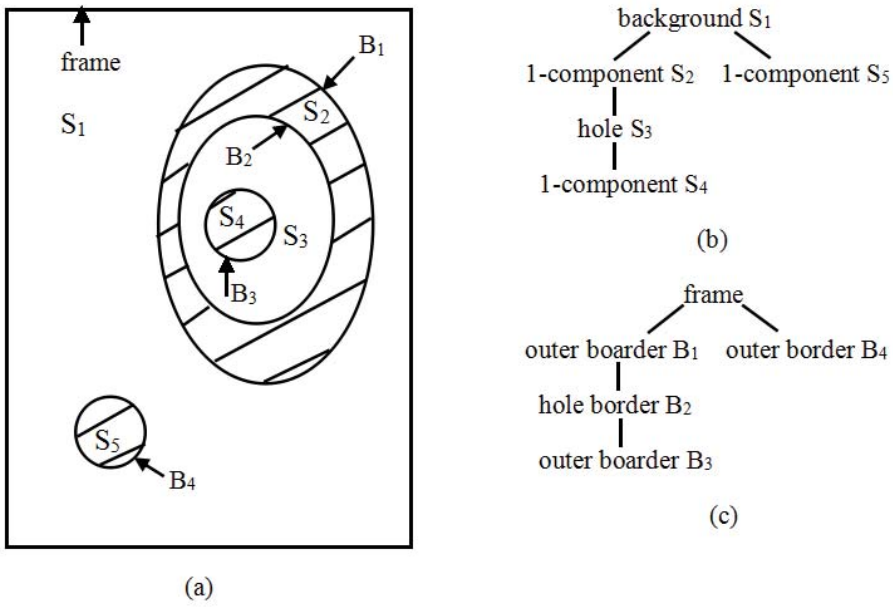

Figure 4.4: An example of surroundness and border [71]

Figure 4.4 is an example of surroundness and border. Surroundness among borders: if there is a sequence of borders like B_0, B_1, B_2, B_3, ..., B_{n-1}, B_n, in which B_1 is the parent border of B_0, B_2 is the parent border of B_1,..., and B_n is the parent border of B_{n-1}. We can say B_n surrounds B_0.

- **Algorithm**

The process is to find all pixels which meet the requirements as border following starting point and mark the pixels on the border. The conditions are shown in Figure 4.5: (a) is the condition for outer border and (b) is for hole border. Figure 4.6 is the rule to determine its parent border.

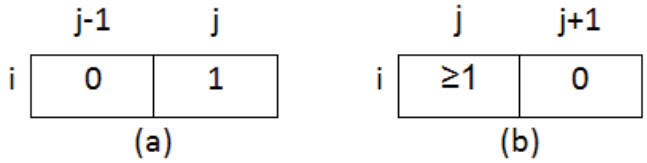

Figure 4.5: Conditions for outer and hole borders [71]

Type of the border B' with the sequential number LNBD / Type of B	Outer border	Hole border
Outer border	The parent border of The border B'	The border B'
Hole border	The border B'	The parent border of the border B'

Figure 4.6: Decision rule for the parent border [71]

The process of the algorithm can be explained through a simple example. For given a frame in Figure 4.7 (a), the circled 1-pixel which satisfies the requirement as a starting point of an outer border is found during scan. It can be marked using sequential number 2. The next step is to change all the pixel values of this border to 2 or -2 as shown in (b). Then scan resumes from circled pixel in (b) until reaching the circled pixel in (c). Clearly, circled pixel in (b) is a starting point of a hole border while the circled pixel in (c) is a parent border of number 2 border. According to marking policy, pixels in this border are changed to 3 or -3. Repeat these steps and we can get the result shown in (e). In Figure 4.7, ob is outer border and hb means hole border

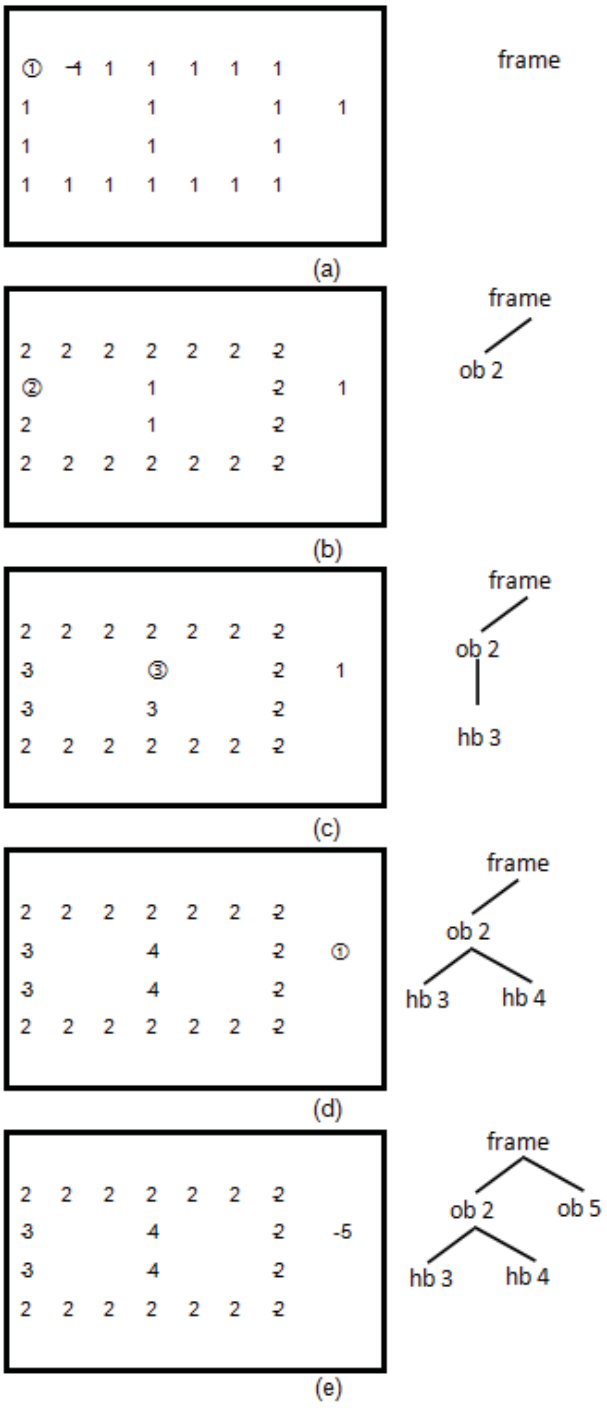

Figure 4.7: The process of border following algorithm [71]

There are five advantages of this algorithm:

1. All 0-components and 1-components can be marked in a binary digital image.

2. Using the border following starting point of its outer (or hole) border, every 1-component (or 0-component) can be denoted.

3. 1-components and holes can be sorted based on a threshold.

4. The surroundness among connected components can be represented in terms of a sequence of border which shows the parent border of every border.

5. We can analyze the topological structure of image which is very important in storing and processing.

4.1.1.2 Convex Hull Algorithm

The convex hull of a polygon represents the original features and is easier to analyze. The contours of gesture are usually simple polygons. In this thesis, we use R. L. Graham's [72] algorithm to find the convex hull which is suitable for simple polygon.

- **Basic Concepts**

(a) How to present a simple polygon

A simple polygon P can be represented using the vertices forming its boundary. It is denoted in terms of $P=<v_1, v_2, v_3,...v_{n-1},v_n>$ whose elements are all the vertex and followed by clockwise orientation.

(b) Path for points

Given two random points on P, the path between them is written as $P[v_i, v_j]$ and the orientation is the same as P. For two paths p and q, the concatenation between them is represented as p o q. For two vertex a and b, if c is in the right or left half-plane of the O[a,b] (the path between a and b), it is defined c lies to the right or left of O[a,b] [72].

(c) Convex hull

All the vertices which have the extreme coordinates form the convex hull of polygon P. It is represented as H(P). Convex hull must meet two conditions: $<v_1, v_2, v_3,...v_{n-1},v_n>$ forms a convex polygon; every point of $L[v_{i-1},v_i]$ lies to the right of $L[v_{n-1},v_n]$ when $2 \leqslant i \leqslant n+1$ [73]. Conditions are shown in Figure 4.8.

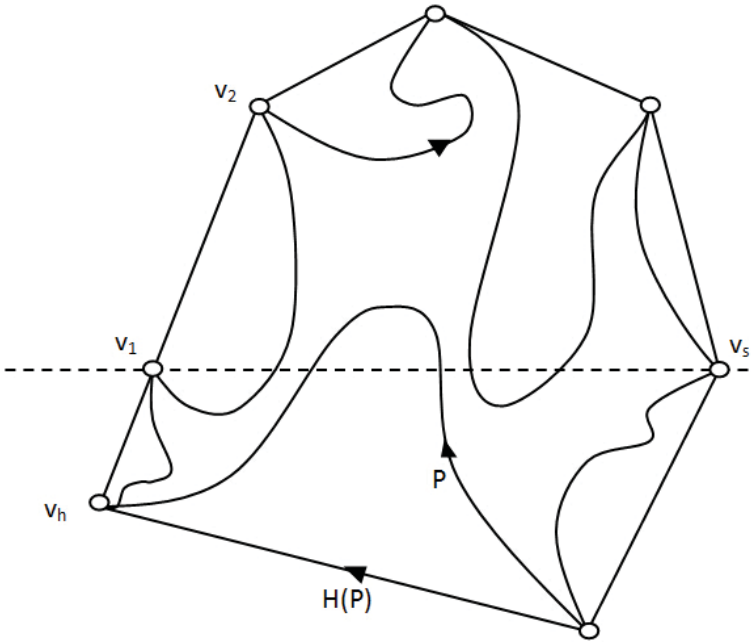

Figure 4.8: Convex hull of a simple polygon [73]

■ Algorithm Lefthull

For a given polygon $P[v_1, v_m]$, its vertices are analyzed in the order $<v_m,v_1,...v_{m-2},v_{m-1}>$. Assume a stack $Q=<q_0,q_1,...q_{t-1},q_t>$ which q_t represents the top of the stack while q_0 represents the stack bottom. The input vertex is denoted by variable x and the immediately preceding stack bottom is denoted by variable y. The algorithm is shown in Figure 4.9.

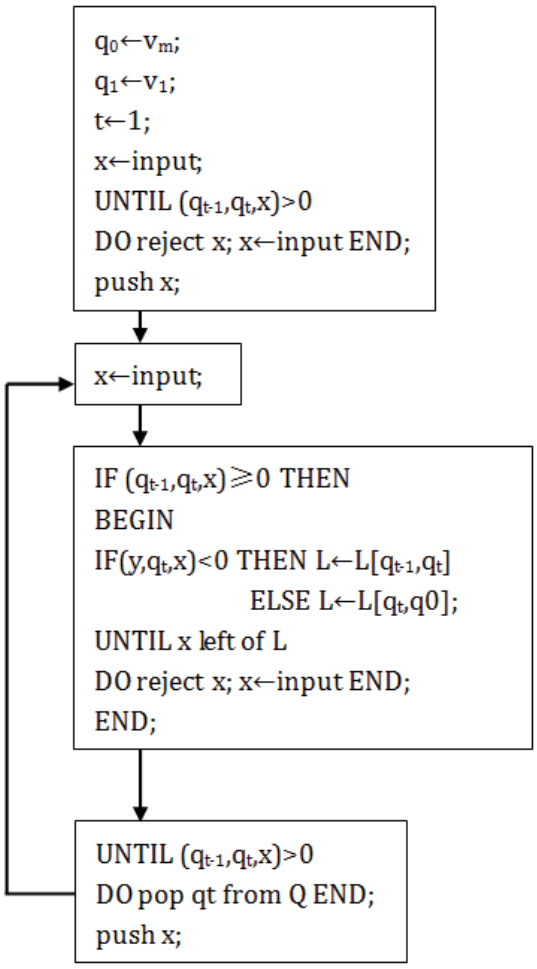

Figure 4.9: Algorithm lefthull [73]

4.1.1.3 Ramer-Douglas-Peucker Algorithm

Ramer-Douglas-Peucker [74] algorithm is also called iterative end-point fit algorithm. It is used to simplify a linear curve by reducing the amount of points. In this thesis, we use this algorithm to simplify the convex hull. To determine whether a point is reduced or not is based on the distance between it and the simplified curve.

An example of using Ramer-Douglas-Peucker algorithm is shown in Figure 4.10. First make a line using the endpoints p_1 and p_8. Then analyze the distance between every point and line x. Keep the furthest point p_3 and analyze the point between them. As point p_2 is not further than the line y to line x, discard it and keep

analyzing the rest points. Point p_5 is the furthest to line z, so keep it and analyze the point p_4. Keep repeating the steps until it reaches the end point.

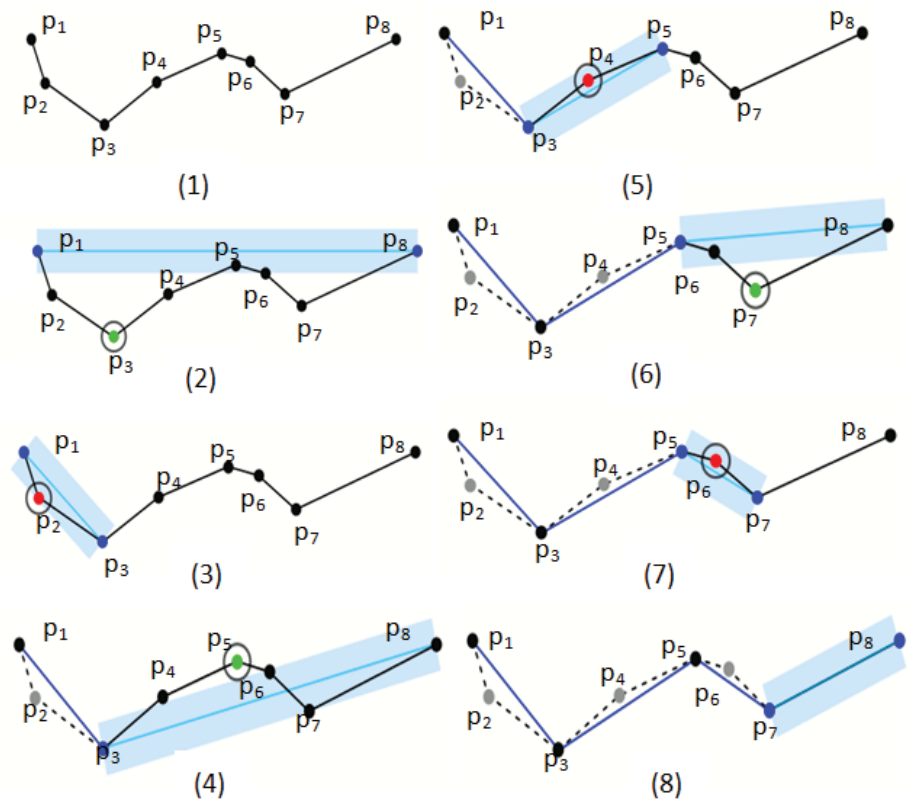

Figure 4.10: An example of using the algorithm [74]

4.1.2 Voice Control Algorithm

DTW algorithm is applied for voice recognition. It is often used in isolated speech recognition. It can measure the similarity between two time series which may vary in time or speed [45].

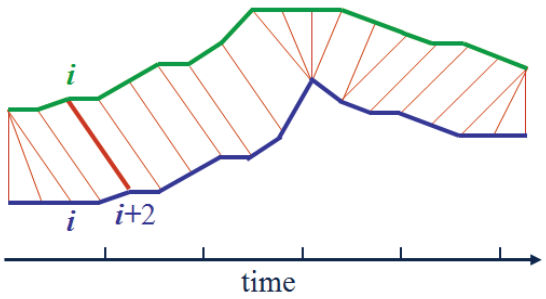

Figure 4.11: The optimal alignment for the given two sequences [94]

For given two sequences A := (a_1,a_2, ..., a_n), B := (b_1, b_2,..., b_m), the optimal alignment can be found under certain restrictions [94]. The time alignment of the two sequences are shown in Figure 4.11. The red lines indicate the aligned points.

To find the optimal alignment of the two sequences, it is essential to build the wrapping path. Wrapping path is also known as alignment path which is a sequence of points. For the given sequences A and B, the wrapping path is P := (p_1, p_2,..., p_k), shown in Figure 4.12.

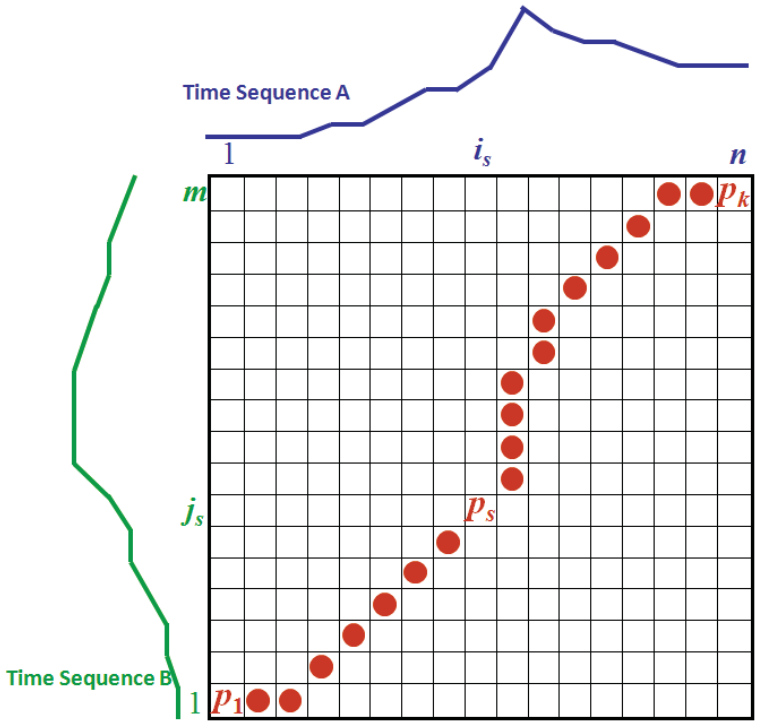

Figure 4.12: Wrapping path between the two sequences [94]

The points of path must meet five restrictions [94]:

1. Monotonic condition: $i_{s-1} \leqslant i_s$ and $j_{s-1} \leqslant j_s$. It means the indexes (i and j) never decrease. This condition avoids the repeat of features in the alignment.
2. Continuity condition: $i_s - i_{s-1} \leqslant 1$ and $j_s - j_{s-1} \leqslant 1$. It means the maximum increase for each step of the indexes is 1. This condition avoids the omission of important features.

3. Boundary condition:$i_1 = 1$, $i_k = n$ and $j_1 = 1$, $j_k = m$. It means the path start and end points must be the bottom left and the top right. This condition assures the complete sequences are analyzed.
4. Warping window condition:$|i_s - j_s| \leqslant r$, where $r > 0$ is the window length. It means the points of path should not wander too far. This condition avoids the omission of different features and repeat of similar features.
5. Slope constraint condition:$(j_{sp} - j_{s0}) / (i_{sp} - i_{s0}) \leqslant p$ and $(i_{sq} - i_{s0}) / (j_{sq} - j_{s0}) \leqslant q$, where q/p is the numbers of steps in the x/ y-direction. It means the slope of path should not be too big or too small. This condition avoids the match of very short parts to relatively very long ones.

4.2 The Embedded System Platform

4.2.1 Profile of Embedded System

4.2.1.1 Characteristics

There are several typical characteristics of embedded system in general:

- Dependable: it must be reliable, maintainable, and safe.
- Most embedded systems are also real-time systems: this kind of systems must guarantee timeliness in system response.
- Efficient: it involves energy efficient, code-size efficient, weight efficient, cost efficient and run-time efficient [75].
- An embedded system is typical a reactive system which is continual interaction with environment and executes [75].
- An embedded system is often connected to physical environment through peripheral hardware like sensors [75].
- It is typical a hybrid system which contains analog and digital parts.

4.2.1.2 Hardware information

The embedded system hardware used in this thesis is FriendlyARM Mini210s development board, which is based on ARM Cortex-A8 processor. The parameters of the main hardware are shown in Table 4.1:

Table 4.1: Hardware information for Mini210s [78]

Hardware	Parameters
CPU	1 GHz Samsung S5PV210 with PowerVR SGX540 graphics engine
RAM	512 MB, 32 bit Bus
Flash	1 GB on board Flash
Ethernet	RJ-45 10/100M (DM9000)
LCD Interface	24 pin (2.0 mm) MIPI and 41 pin connector for Displays
Serial Ports	1x DB9 connector (RS232), total: 4x serial port connectors
Debug	10 pin JTAG (2.0 mm)
Expansion	40 pin GPIO, 20 pin SDIO (SD, SPI, I2C), 20 pin Buttons/Keypad (2.0 mm)
Power	regulated 5V (DC-Plug)
User Inputs	4x push buttons and 1x A/D pot
User Outputs	4x LEDs
OS Support	Windows CE 6 Linux Android

The development board is shown in Figure 4.13. Base on the richness of I/O connection possibility of this development board, a series of other peripheral equipment are attached on the development board. The selection and build of those equipment will be introduced in the next section 4.3. Those peripheral equipment include USB camera [79], voice recognition module [80], ZigBee module [81], relay [62] and devices [83, 84].

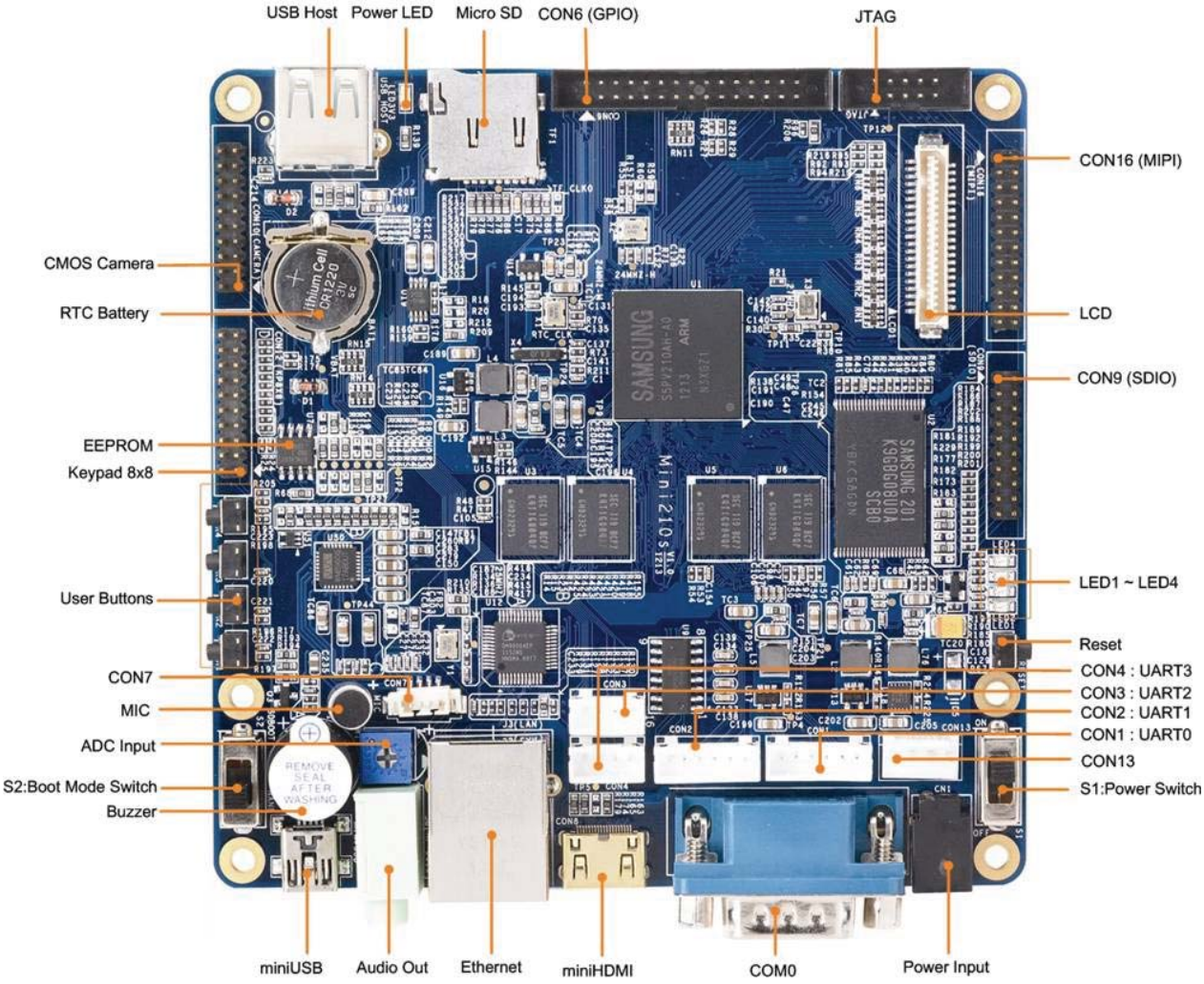

Figure 4.13:The development board [78]

4.2.1.3 Embedded operating system

Embedded Operating System is optimized for embedded computing systems which is normally constrained by limited resources such as limited RAM and ROM. In this thesis, Embedded Linux is used for the development board.

4.2.1.4 Composition of embedded Linux software system

For typical embedded Linux system, there are 4 layouts as Figure 4.14 shows, being known as Bootloader, Linux kernel, root file system and upgrade space.

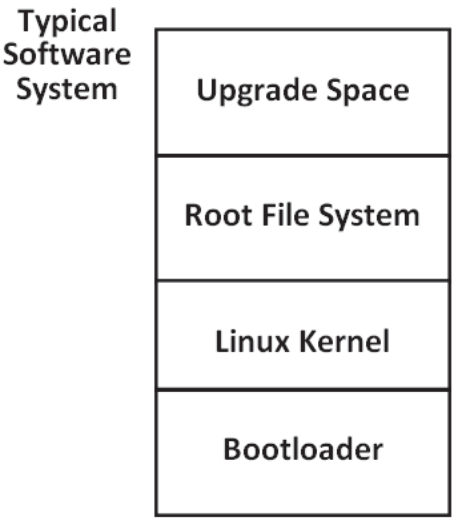

Figure 4.14: Typical flash memory layout [68]

Bootloader is the first section of code which is executed after powering up the embedded system. Bootloader can be regarded as the bottom of the whole system. After bootloader is initialized, Linux kernel will be processed in the response of memory management and resource management. On the top of kernel, a root file system is essential to run the operating system after compiling the kernel. Upgrade space is presented to create specified applications and functions for users. In this project, embedded Qt [85] has been applied for further coding and developing. Figure 4.15 shows the basic framework of Qt for Embedded Linux. Qt ("cute") is a cross-platform application and UI development framework for multi-systems [65], which can be applied in embedded Linux system. Qt uses standard C++ and provides both GUI and non-GUI support for developing.

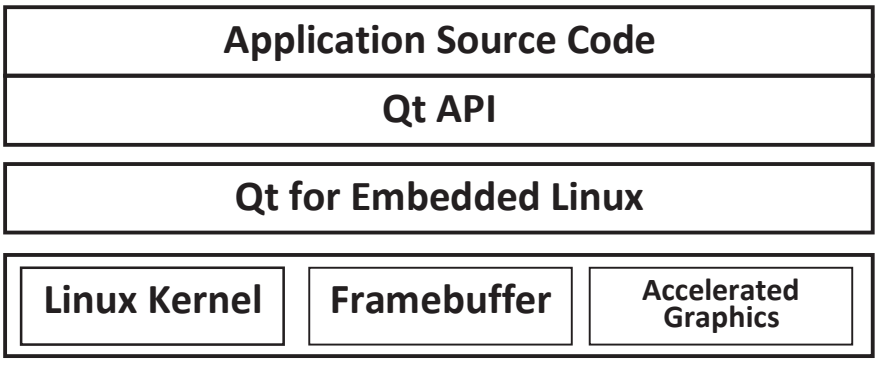

Figure 4.15: Framework of Qt for Embedded Linux [77]

4.2.2 The Development Board System

4.2.2.1 Linux kernel

Linux kernel is the most important part of the system, and a bunch of changes should be made to fit the selected hardware so that the porting can be accomplished into the target board.

The contents and illustrations of Linux kernel source is listed in Table 4.2:

Table 4.2: Main menus and illustration of Linux kernel source

Linux kernel Main menu and illustration	
arch	contains instructions which relates to CPU architectures, such as i386 arm
include	contains Linux kernel instructions and header files
init	contains initialize codes of Linux kernel which is the beginning of working
drivers	contains device drivers in Linux kernel
fs	contains all codes for file system
net	contains codes relate to networking
mm	contains codes for memory management
ipc	contains codes relate to inter-process communication
kernel	contains main kernel codes, including process, execution, signal and so on

With the information from the hardware the of development board, a series of modification to drivers, under 'driver' menu from the list, should be carried out first before the Linux building. The kernel source can be configured and then compiled using the following commands:

```
#make ARCH=arm CROSS_COMPILE=arm-linux-
#make s5pv210_config
#makemenuconfig
#make
```

The menuconfig interface is shown in Figure 4.16, drivers and hardware settings will be configured and saved through this interface.

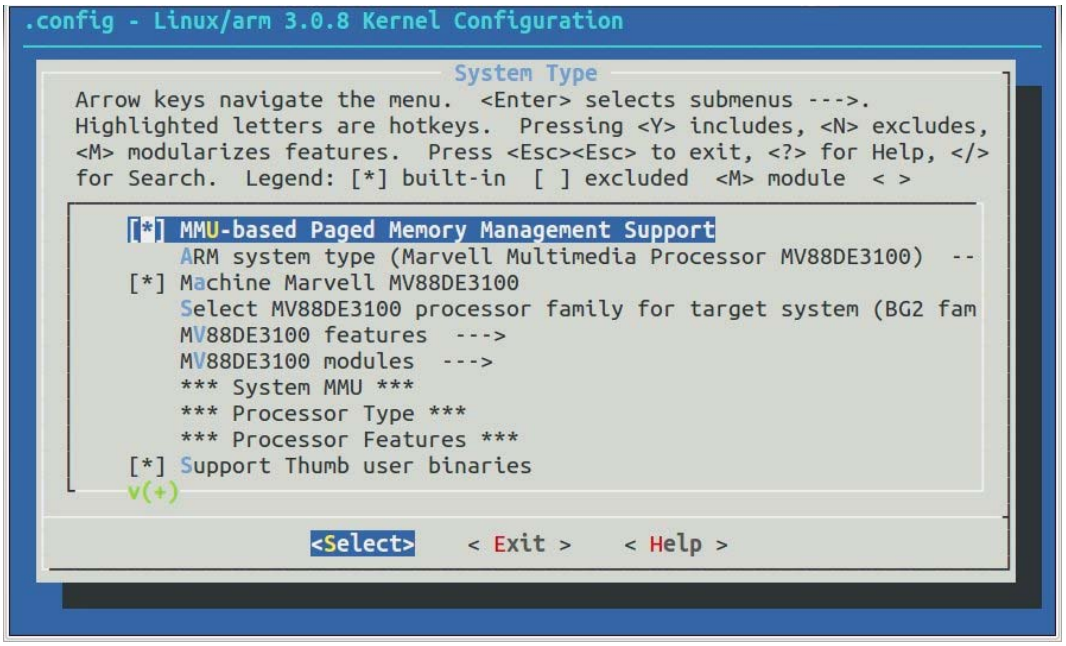

Figure 4.16: Linux kernel configuration

Linux kernel image, a zImage will be generated after compiling, which then can be downloaded into target board.

4.2.2.2 Root file system

File system is a management system to manage and store file information, which contains three parts: software relates to file management, files to be managed and data structure required by file management.

There are different types of file system format, such as Cramfs, JFFS and YAFFS. In this thesis YAFFS2 is used as the file system format.

4.2.3 Cross Compilation Environment

As the limited CPU processing speed and RAM size, the development of software/applications would be unable to carry out in the embedded system. There should be a faster PC host system to develop and debug the applications. In other

words, applications are developed in host system and then downloaded to development board.

The architectures of PC host system and embedded target system are different, so in order to let the software developed in PC system to work in embedded system, a cross-compilation environment need to be constructed. The construction of cross compilation environment contains PC host system, cross compiler, matched software environment and the communications between host system and target board.

4.2.3.1 Linux host system

As a host PC for this thesis, a lap-top powered by Core™ i7 4500U processor with Ubuntu 13.04 is used. In order to accomplish the cross compilation, a series of settings and applications need to be installed as following.

a) Minicom

Minicom is a tool for serial port connection, which can be installed by typing the following command in the terminal.

```
#sudo apt-get installminicom
```

As there is no RS232 port in the laptop, a PL2303 USB to RS232 convertor is used for the serial port connection. No driver need to be installed since Ubuntu provides the PL2303 driver by default. Type the following command in the terminal to check the serial port information. As Figure 4.17 shows, the PL2303 converter is attached to the USB.

```
#dmesg | grepttyUSB
```

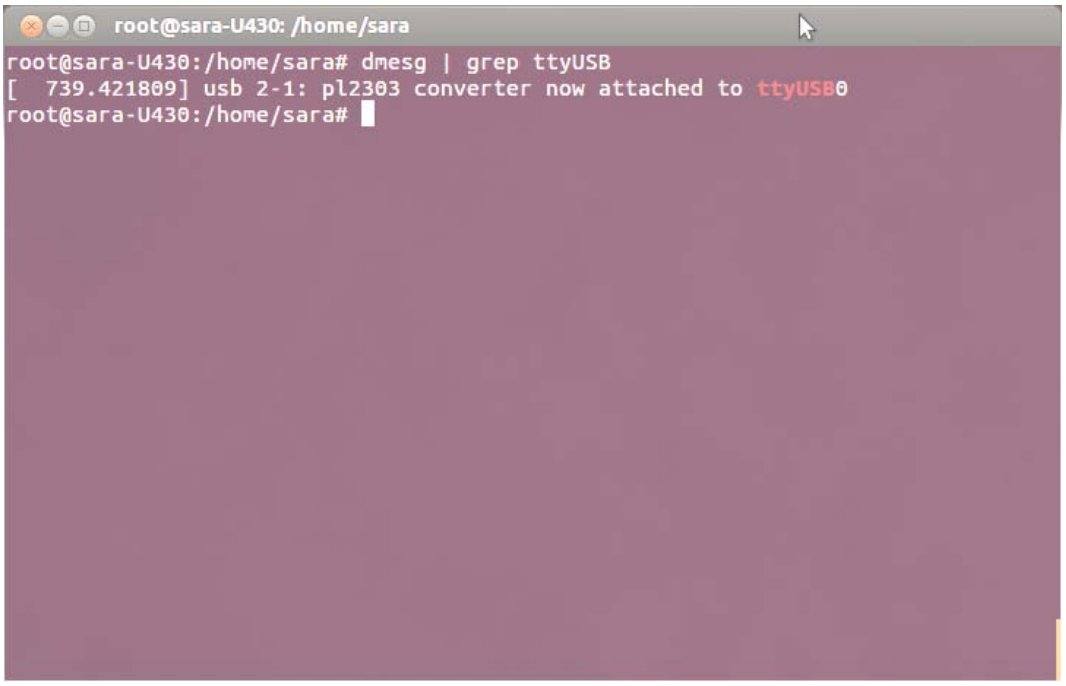

Figure 4.17: Step 1 of minicom setting

A modification to the setting page of minicom (shown in Figure 4.18)is carried out based on the information from the previous check.

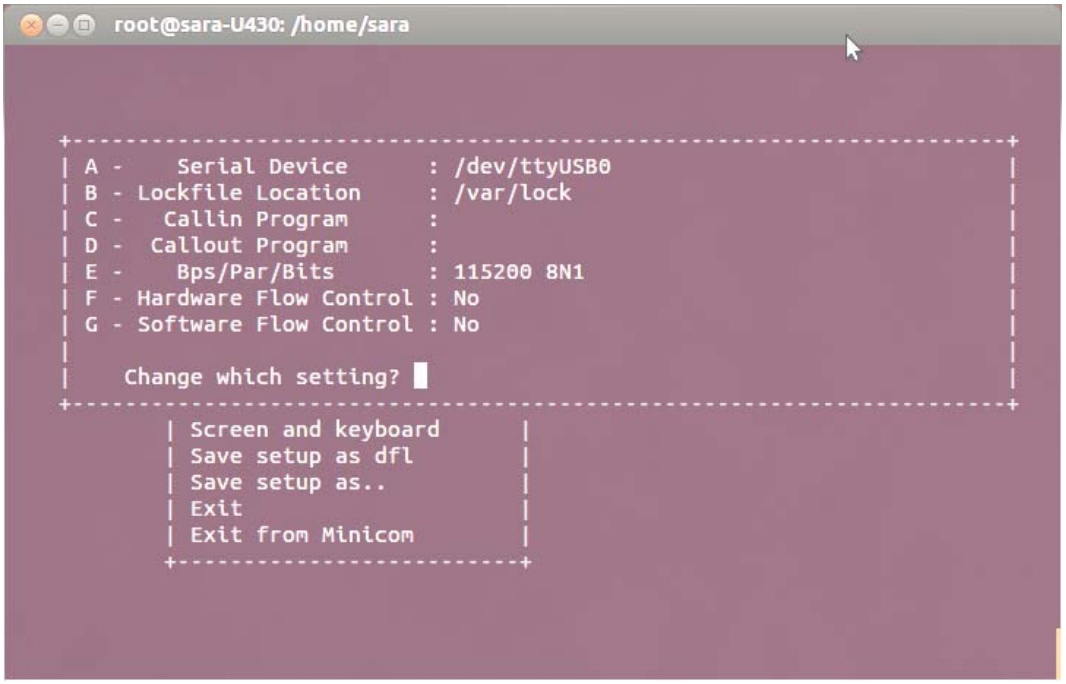

Figure 4.18: Step 2 of minicom setting

b) NFS kernel

Network File System (NFS) is a distributed file system and it is developed by Sun Microsystems in 1984 [86]. It offers the data sharing by different machines and systems via network. Through NFS, target board can be accessed to certain direction on host system.

Firstly, a installation should be checked by the following command in the terminal.

```
#sudo apt-get installnfs-kernel-server
```

After that, information of NFS sharing direction should be added to the end of file "/etc/exports" as below.

```
/home/sara/nfs *(rw, sync, no_root_squash)
```

The first is to assign sharing content and the * symbol indicates the accessible IP address. And the following parameters mean:
rw – read/write permission
sync – synchronize data into ram and disc
no_root_squash – this parameter is to request server to permit client system reach contents by root level permission.

With modification of the settings, restart the NFS server (shown in Figure 4.19) and the PC now is ready to provide share as NFS host.

```
#sudo /etc/init.d/nfs-kernel-server restart
```

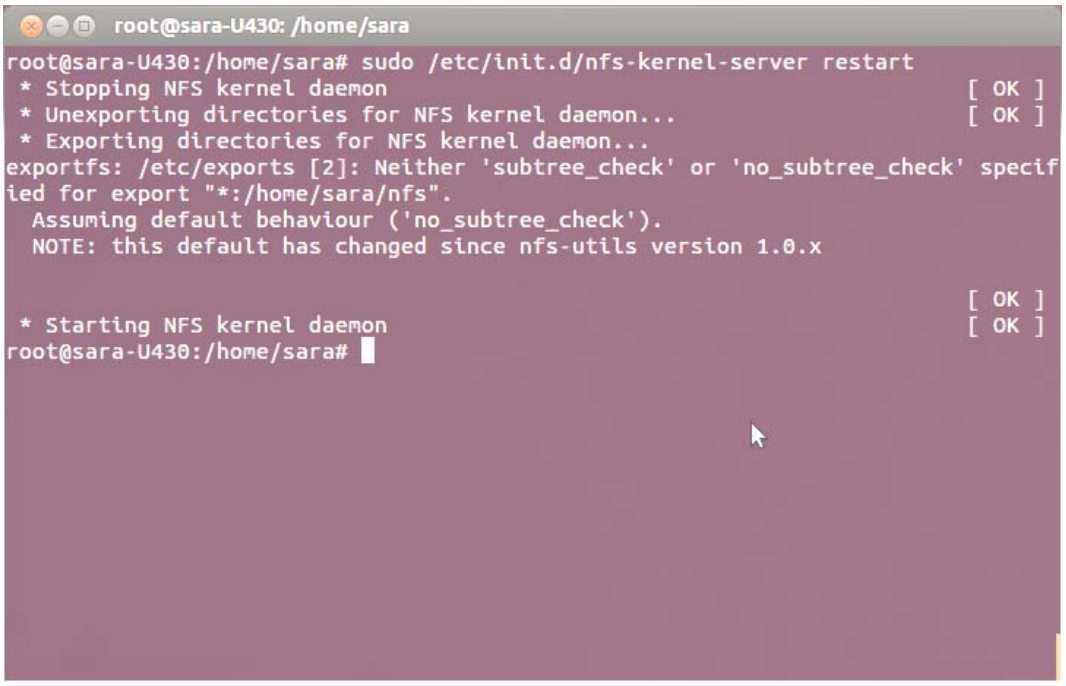

Figure 4.19: Restarting of NFS service

c) Cross compiler (arm-linux-gcc)

A cross compiler can create executable code for a platform which is different with the platform the compiler is running. In this case, a arm-linux-gcc is used as a cross compiler in the host PC. The version of the compiler is arm-linux-gcc 4.5.1.

The very first step is to decompress and install the compiler by typing the following command in the terminal:

```
#sudo tar zxvf arm-linux-4.5.1-v6-vfp.tgz -C /
```

In order to make the compiler to be used as a default cross compiler and to be found later, the following indication information should be added to the file "/etc/environment".

```
:/opt/toolschain/4.5.1/bin
```

To check the installation, the following command can be used in the terminal, which can provide the version information as shown in Figure 4.20.

```
#arm-linux-gcc -v
```

Figure 4.20: Version check for cross compiler

d) Qt

In order to get a matched environment in both host PC and the target board, a Qt 4.8.5 for Linux-X86 is installed in host PC and Qt in the same version for Linux-ARM is cross-compiled and ported to the target board. The cross compilation of Qt for the target board will be introduced in section 5.1.3.3.

A Qt 4.8.5 for Linux open source can be downloaded from *qt-project.org/downloads*. A series of essential packages need to be installed first by following commands. After that, installation of Qt can be processed.

```
#sudo apt-get install build-essential
#sudo apt-get update
```

```
#sudoapt-get install libx11-dev libfreetype6-dev libavahi-gobject-
devlibSM-devlibXrender-devlibfontconfig-devlibXext-dev
#tar zxvf qt-everywhere-opensource-src-4.8.5.tar.gz
#./configure
#make
#make install
```

e) OpenCV

OpenCV is short for Open Source Computer Vision Library, which is a library for computer vision and machine learning. The library contains more than 2500 optimized algorithms, which can be used for face detection, object identification, action classification and so on. The version of OpenCV used in this thesis is OpenCV 2.4.5 for Linux. The OpenCV, equals with Qt, should be installed in both host PC and the target board. The cross compilation of OpenCV for the target board will be introduced in section 5.1.3.4.

Before the installation, some essential packages need to be installed using the following commands in the terminal.

```
#sudo apt-get install build-essential
#sudo apt-get installcmakegit libgtk2.0-dev pkg-configlibavcodec-
devlibavformat-devlibswscale-dev
#sudo apt-get install python-dev python-numpy libtbb2 libtbb-
devlibjpeg-devlibpng-devlibtiff-devlibjasper-dev libdc1394-22-dev
```

OpenCV can be downloaded from *github.com/Itseez/opencv/tree/2.4.5*, and the installation can be done using the following commands.

```
#cd ~/opencv
#mkdir release
#cd release
#cmake -D CMAKE_BUILD_TYPE=RELEASE -D
CMAKE_INSTALL_PREFIX=/usr/local ..
#make
#sudo make install
```

4.2.3.2 Target board system

In order to communicate to the host PC through NFS, the target board should be connected with host PC using RJ45 wire. And then set the IP address of target board to the same network segment, for example, the IP address of host PC is 192.168.0.11 and the target board is 192.168.0.100. And then the configuration should be done using the following commands in the terminal.

```
#mkdir /mnt/nfs
#mount –t nfs –o nolock,intr,rsize=4096,wsize=4096
192.168.0.11<host IP address>:/home/sara/nfs /mnt/nfs
```

After the set above, the target board could access the contents on host system through */mnt/nfs*.

4.2.3.3 Qt cross compilation and porting

The same version of Qt should be cross compiled for target board, and the following configurations are done. After the installation of Qt, the libraries and fonts of the installed Qt files is ported into target board using NFS.

1. Modify in *<Qt Source>/mkspecs/common/g++.conf*

```
QMAKE_CFLAGS_RELEASE    += -O0
```

2. After the modification above, switch to Qt source direction and use configuration instruction;

```
#./configure  -prefix /opt/qt-4.8.5-arm  -release -shared -fast -no-
largefile -qt-sql-sqlite -no-qt3support -no-openssl -xplatformqws/linux-arm-
g++ -embedded arm -little-endian -no-mouse-linuxtp -qt-mouse-tslib -
I/usr/local/tslib/inclued -L/usr/local/tslib/lib
```

3. Copy generated Qt library and fonts into target board via NFS.

4.2.3.4 OpenCV cross compilation and porting

As well as Qt, the OpenCV for target board is installed in the same version and ported into target board, and a cmake 2.8.7 is needed in the compilation, which can be downloaded from *cmake.org/files/v2.8/*. The configuration and compilation are shown below.

1. Open GUI interface of cmake by typing in the terminal;

```
#cmake-gui
```

2. Choose directions for source code and binaries, then click *Configure* and remain *Unix Makefiles*, and choose the last one to specify *options for cross-compiling*, as shown in Figure 4.21;

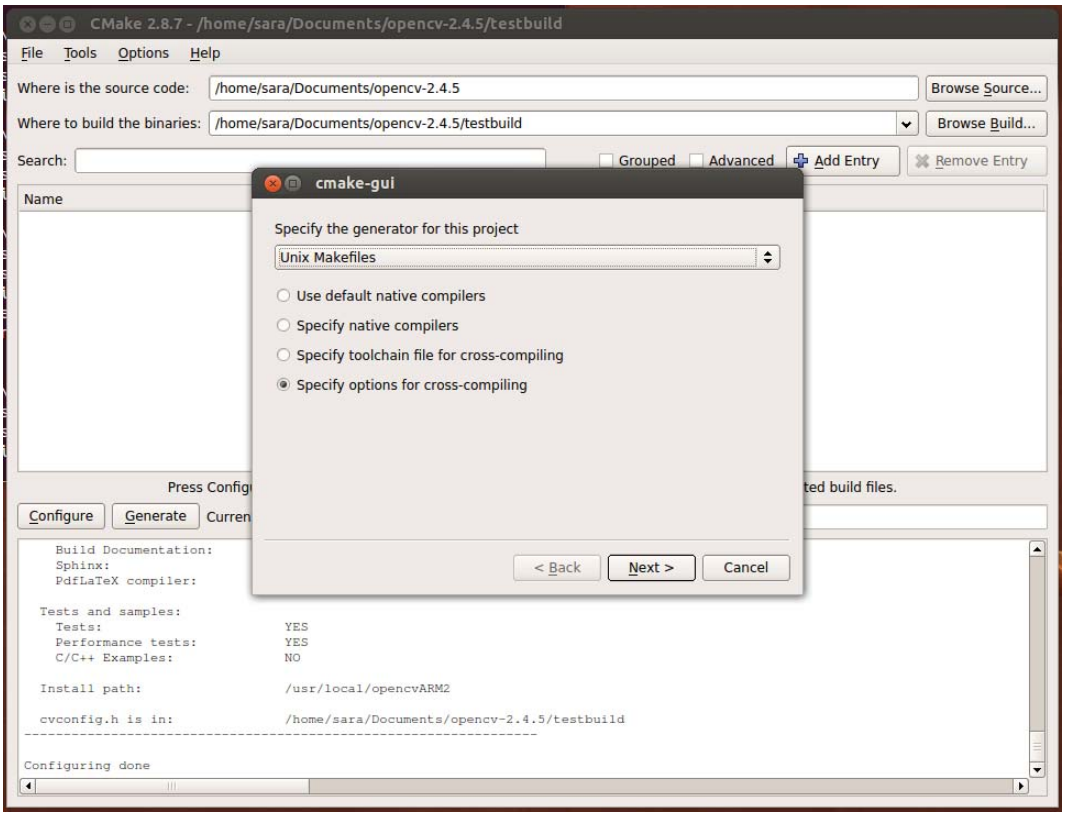

Figure 4.21: CMake configuration for OpenCV cross compilation – configure step2

3. Type in *target system* information, choose *compilers* and *target root*, then *finish*, as shown in Figure 4.22;

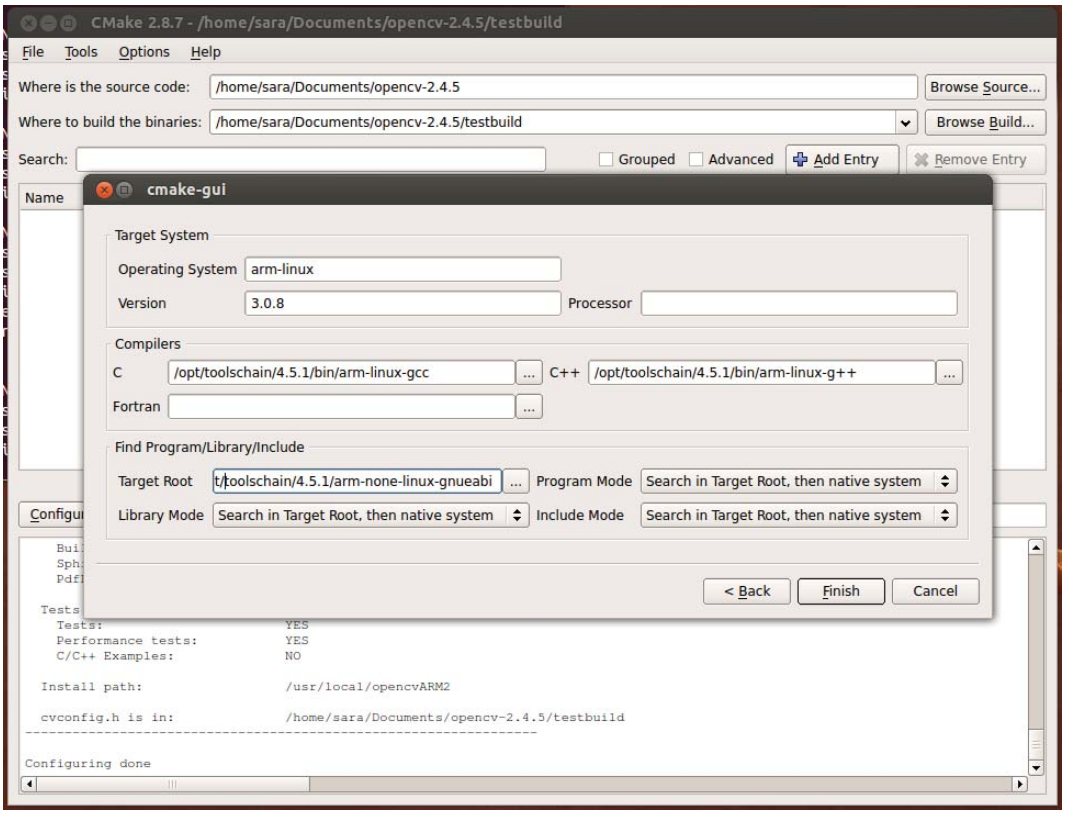

Figure 4.22: CMake configuration for OpenCV cross compilation – configure step3

4. Modify some configures in the GUI interface, and then *generate*;

WITH_TIFF	OFF
PREFIX	/opt/toolchains/opencv244arm/
CMAKE_EXE_LINKER_FLAGS	-lpthread–lrt

5. Run make and make install in the binaries direction selected in step2;

```
#make
#make install
```

6. Copy generated OpenCV libraries and includes into target board via NFS.

4.2.4 Summary to the Embedded System Platform

The embedded system is the base of application development and the foundation of the whole system design. A cross compilation environment with host PC is introduced in this project.

For the target board, an embedded Linux kernel is installed and bootstrapped by bootloader. And then Qt and OpenCV would be cross compiled and ported into target board.

The function of host system is to compile and prepare files for target board and provide an environment to code and debug easily. Due to the different environment between host system and target board, all the source files and application code would be cross-compiled and then transferred into board via NFS connection.

4.3 Hardware Design

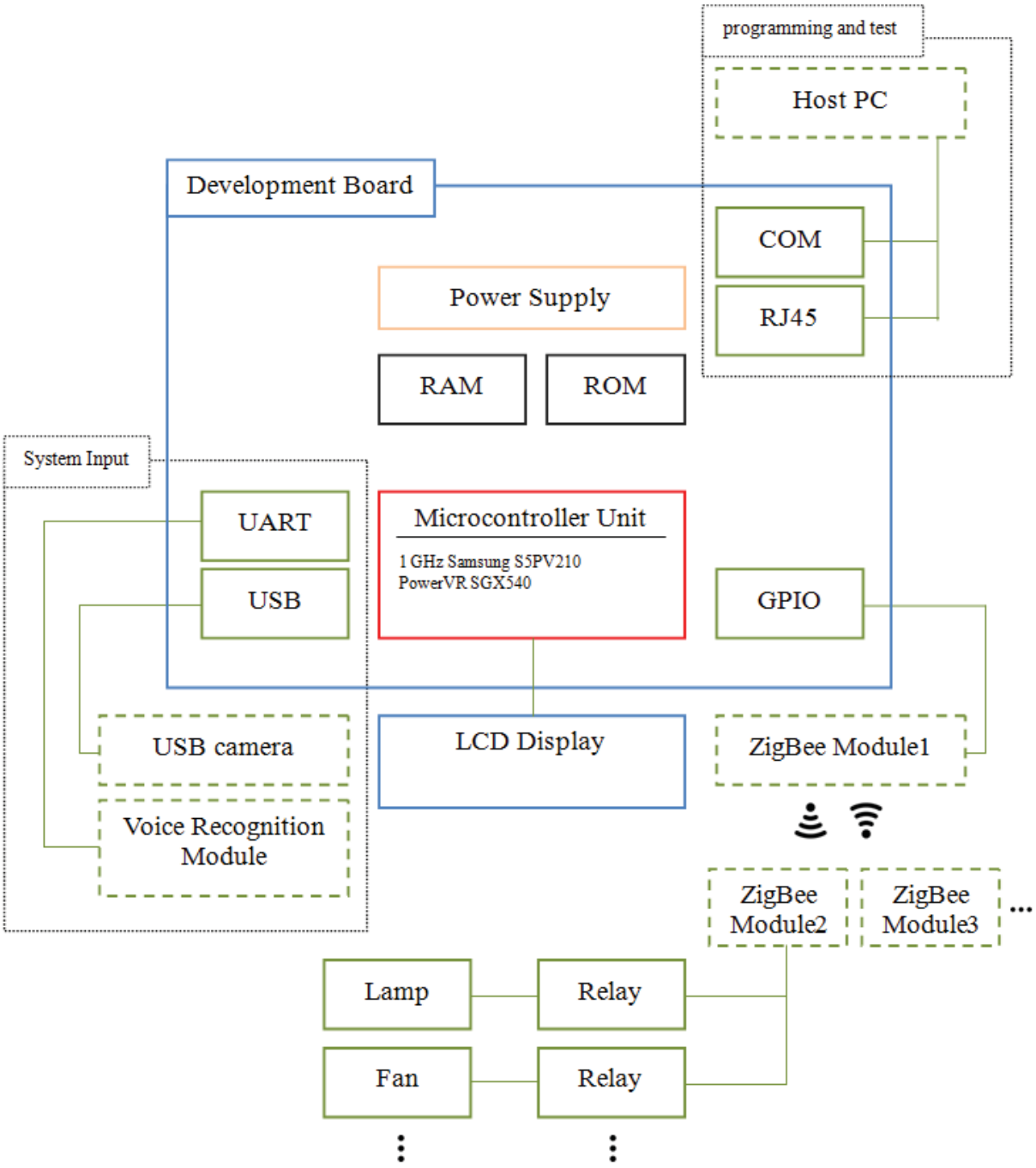

Figure 4.23: Hardware overview

The system hardware is shown as Figure 4.23. With the rich connection ability of the development board, different peripherals and modules are connected into the

system via different interfaces. A camera is connected to the development board via USB interface as gesture input device while a voice recognition module is connected via UART as a voice input device. A ZigBee module is connected to the board via GPIO interface as a wireless signal transmitter and the other ZigBee module is used as a wireless signal receiver to send signal to relays.

The hardware that used in the system, other than the development board, are introduced: wireless connection module, camera, voice recognition module, relays and devices that under control.

4.3.1 Wireless Connection Module

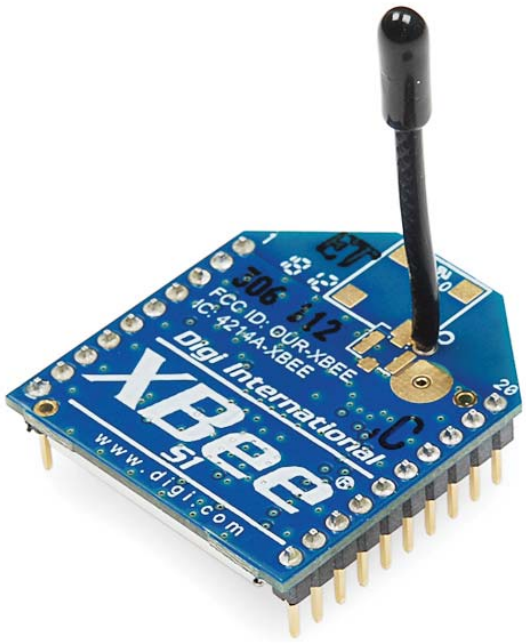

Figure 4.24: Zigbee module [81]

ZigBee is used in this system as the wireless connection method and the ZigBee module used in this system is XBee 802.15.4 which is produced by Digi International Inc, as shown in Figure 4.24. It is a widely used ZigBee module and it is suitable for small project due to its simplicity of configuration. It uses IEEE 802.15.4 networking protocol which can provide efficient point-to-multipoint or peer-to-peer networking [81]. In this thesis, two ZigBee modules are employed: ZigBee module 1 is to receive signal from development board and send signal to

ZigBee module 2; ZigBee module 2 is to receive signal from ZigBee module 1 and send signal to connected relay to control corresponding devices.

4.3.2 Camera

A 10moons V804 camera (left of Figure 4.25) with 640x480 video capture resolutions at 30 fps is attached on the system via USB connection [79].

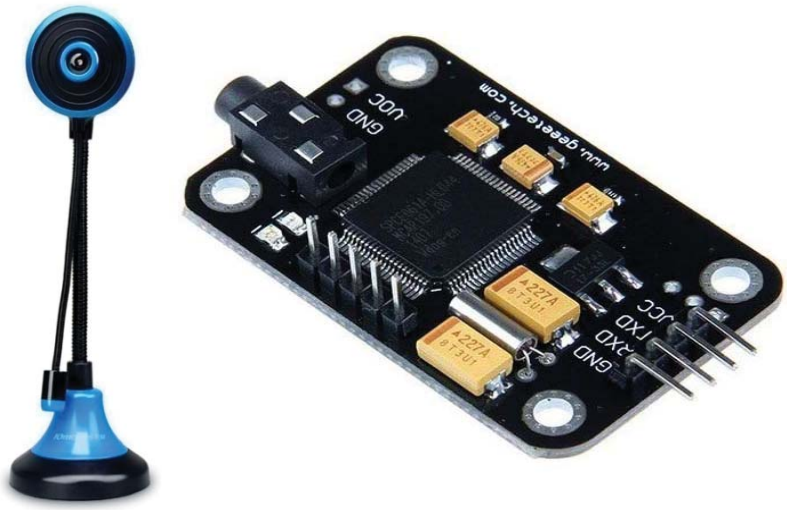

Figure 4.25: 10moons V804 camera and Voice recognition module [79, 80]

4.3.3 Voice Recognition Module

The module is shown in the right of Figure 4.25. It can receive commands or send data through serial port interface. It can restore 3 groups and each group consists of 5 voice instruction. After recording all voice instruction we need, it is ready to recognize voice through microphone when it is in recognition mode. It can recognize commands with high accuracy.

4.3.4 Relay and Devices

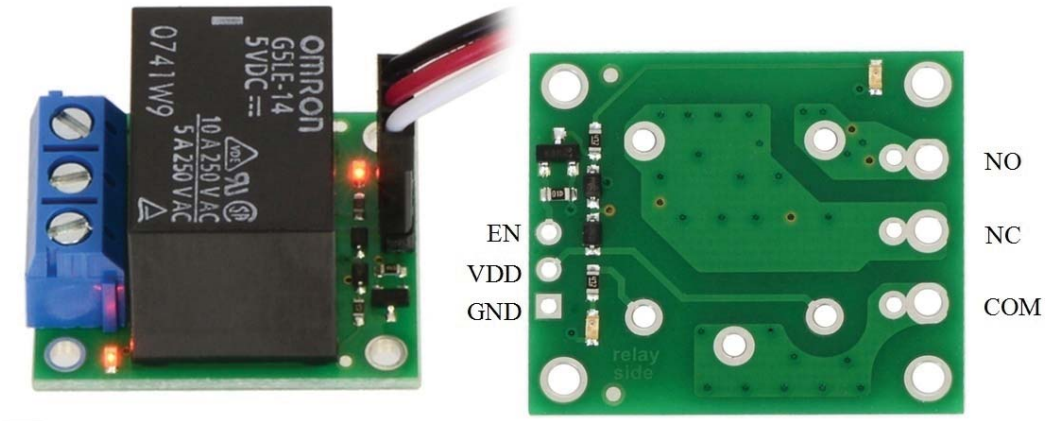

Figure 4.26: Pololu basic SPDT relay carrier[82]

The relay is Pololu Basic SPDT Relay Carrier whose power relay is an Omron G5LE-14-DC5, as shown in Figure 4.26. It can be regarded as a SPDT (single-pole, double-throw) switch and can be controlled by low-voltage and current signals [82]. There are three terminal blocks: EN, VDD, GND for switch connection and three mail header NO (normally open), NC (normally closed), COM (common).

Two devices are chosen as a example of the appliances that can be controlled: desk fan and lamp, shown in Figure 4.27.

The desk fan used in this thesis is a very common fan: Micasa 6" desk fan [83] with 2 speed dial switch. And the lamp is Mi Casa Caitlin table lamp [84] with an inline switch.

Figure 4.27: Lamp and desk fan used in project [83, 84]

4.4 Software Design

There are five steps of building the software system: 1. recognize gesture from real-time frame; 2. control device through GPIO ports; 3. using voice recognition module to recognize voice commands; 4. using gesture and voice to control the devices; 5. user interface

4.4.1 Recognize Gesture from Frame

The process of recognizing gesture from real-time frame is shown in Figure 4.28. There are five stages: 1. capture frame from camera; 2. get sample color of the user hand; 3. extract hand from frame and present it as binary image; 4. recognize gesture from the binary image; 5. show information of the gesture.

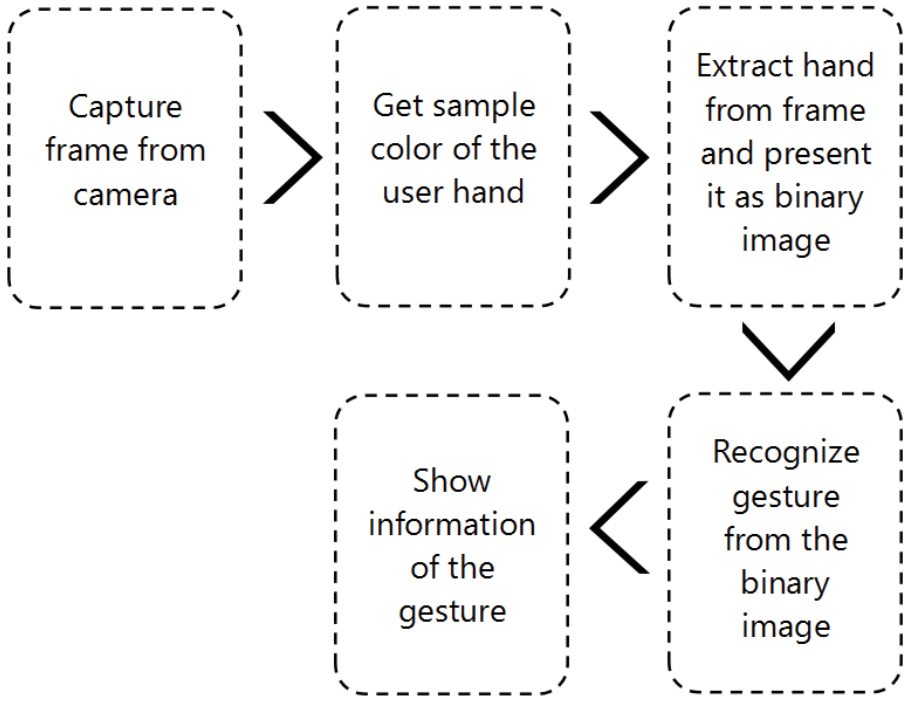

Figure 4.28: The process of gesture recognition

4.4.1.1 Capture frame from camera

V4L2 is employed for image capture in this thesis. It is the second version of V4L which is a Linux kernel API for video capture and other output device like TV tuners [87]. There are three main steps in this stage:

1. Open the device: to open a V4L2 device, the open() function is used with a parameter which represents the IP of device. The key function is [88]:

```
#include <fcntl.h>
int open(const char *path, int oflag, ... );
```

The path in this case is"/dev/video0" which is the path of the USB camera. The flag indicates the access mode.

2. Set format of pixels: the USB camera used in project supports YUV format, so in the corresponding function, the parameter should be V4L2_PIX_FMT_YUYV which tells the application about the pixel format.

3. Get frame from the device: after set the format and parameters, the frame can be captured from the camera. The key function used in this step is:

```
get_frame(unsigned char ** yuv_buffer_pointer, size_t* len)
{
   v4l2_buffer queue_buf;

   queue_buf.type = V4L2_BUF_TYPE_VIDEO_CAPTURE;
   queue_buf.memory = V4L2_MEMORY_MMAP;

   if(ioctl(fd, VIDIOC_DQBUF, &queue_buf) == -1)
   {
      return FALSE;
   }

   *yuv_buffer_pointer = (unsigned char
*)buffers[queue_buf.index].start;
   *len = buffers[queue_buf.index].length;
   index = queue_buf.index;

   return TRUE;

}
```

After calling this function, the YUV buffer from camera frame is obtained and used in the next stage.

4.4.1.2 Get sample color of the user hand

At the beginning of the program, system needs to sample the user's skin color and make a color profile based on it. This color profile can help to extract user's hand from the background.

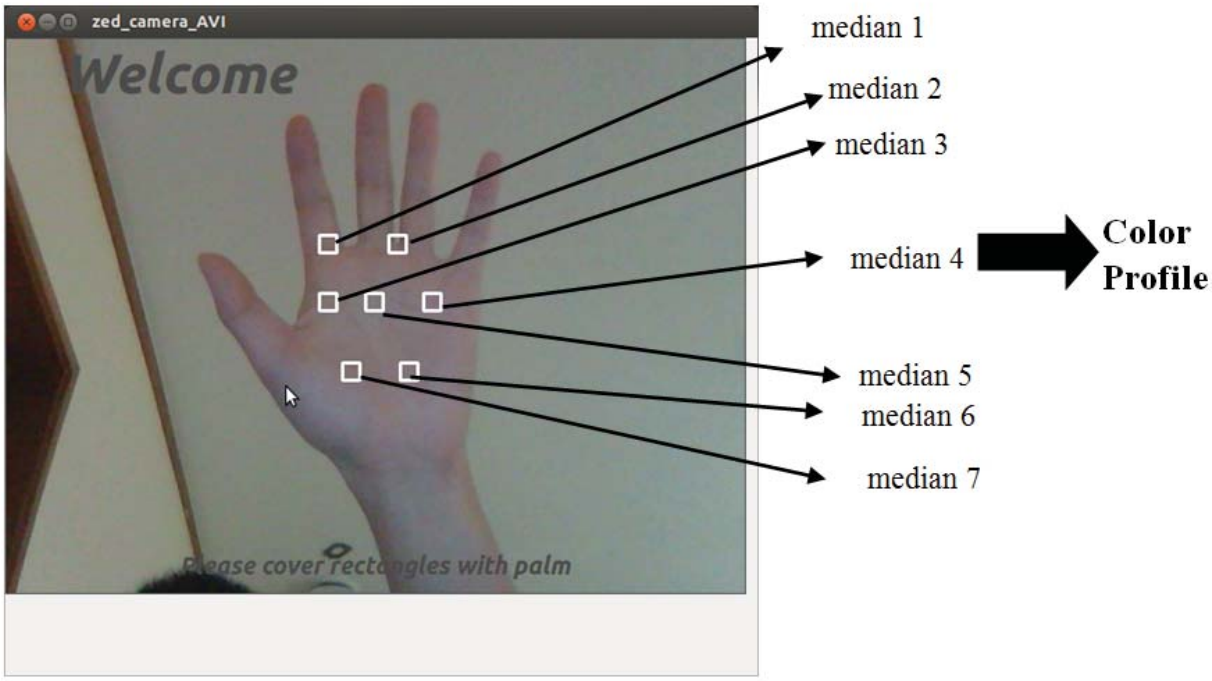

Figure 4.29: Sampling image

The sampling image is shown in Figure 4.29. The system collects the skin color in seven rectangular area. Then analyzes the rectangular area to get the average value. There are two key functions involved: 1. the function to get median of a vector; 2. the function to get the average color of a rectangular mat image. The following function is to calculate the median:

```
Median(vector<int> a){
   int b;
   size_t size = a.size();
   sort(a.begin(), a.end());

   if (size % 2 == 0)
   {
      b = a[size / 2 - 1] ;
   } else{
      b = a[size / 2];
   }

   return b;
}
```

The return value "b" equals the median of vector "a". The function to get average color of a Mat image is:

```
AverageColor(Mat sample,int average[3]){
   vector<int>m0;
   vector<int>m1;
   vector<int>m2;

   for(int i=2; i<sample.rows-2; i++){
      for(int j=2; j<sample.cols-2; j++){


m0.push_back(sample.data[sample.channels()*(sample.cols*i + j) + 0]) ;

m1.push_back(sample.data[sample.channels()*(sample.cols*i + j) + 1]) ;

m2.push_back(sample.data[sample.channels()*(sample.cols*i + j) + 2]) ;
         }
      }
      average[0]=Median(m0);
      average[1]=Median(m1);
      average[2]=Median(m2);
   }
```

Using one of the sampling rectangular image as the input Mat image, we can get the average color in terms of a vector "average[3]".

In this stage, we can get seven vectors correspond to the seven sampling rectangles. Each vector contains three elements.

4.4.1.3 Extract hand from frame and present it as binary image

The hand can be extracted from background by using a threshold. In this case, we sample in seven locations which provides seven color vectors in the profile. Each vector can produce a binary image, so there are seven in total. Sum the seven binary images and filter the noise to get a smooth image. Figure 4.30 shows the process of generating the binary image.

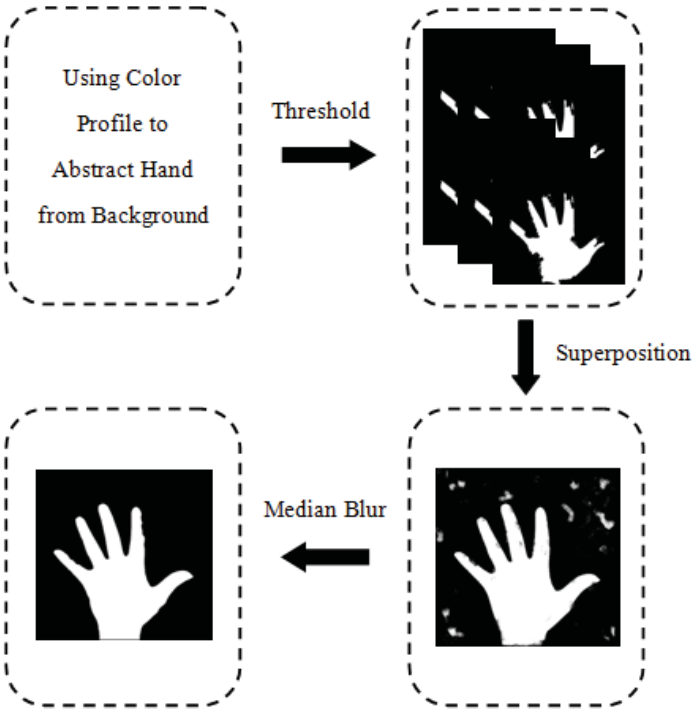

Figure 4.30: The process of generating the binary image

There are two important functions in this step:

1. The function to extract image based on the threshold is [89]:

voidinRange (InputArraysrc, InputArraylowerb, InputArrayupperb, OutputArraydst)

Parameters:

- src - first input array.
- lowerb - inclusive lower boundary array or a scalar.
- upperb - inclusive upper boundary array or a scalar.
- dst - output array of the same size as src and CV_8U type.

It is an OpenCV function. When using the real-time frame as the input image, we can get a binary image based on the threshold. The threshold contains lower boundary array and upper boundary array.

2. The function to use median filter is[82]:

```
void medianBlur(InputArray src, OutputArray dst, int ksize)
```

Parameters:

- src – input 1-, 3-, or 4-channel image; when ksize is 3 or 5, the image depth should beCV_8U, CV_16U, or CV_32F, for larger aperture sizes, it can only be CV_8U.
- dst – destination array of the same size and type as src.
- ksize – aperture linear size; it must be odd and greater than 1, for example: 3, 5, 7...

It is an OpenCV function. It uses a non-linear filter to smooth and denoise the input image.

4.4.1.4 Recognize gesture from the binary image

There are four steps in this stage: 1. find contours in the image; 2. find convex hull of a point set; 3. find points furthest away from convexity defects. 4. discard irrelevant points. Figure 4.31 shows the process of this stage, A is the given binary image.

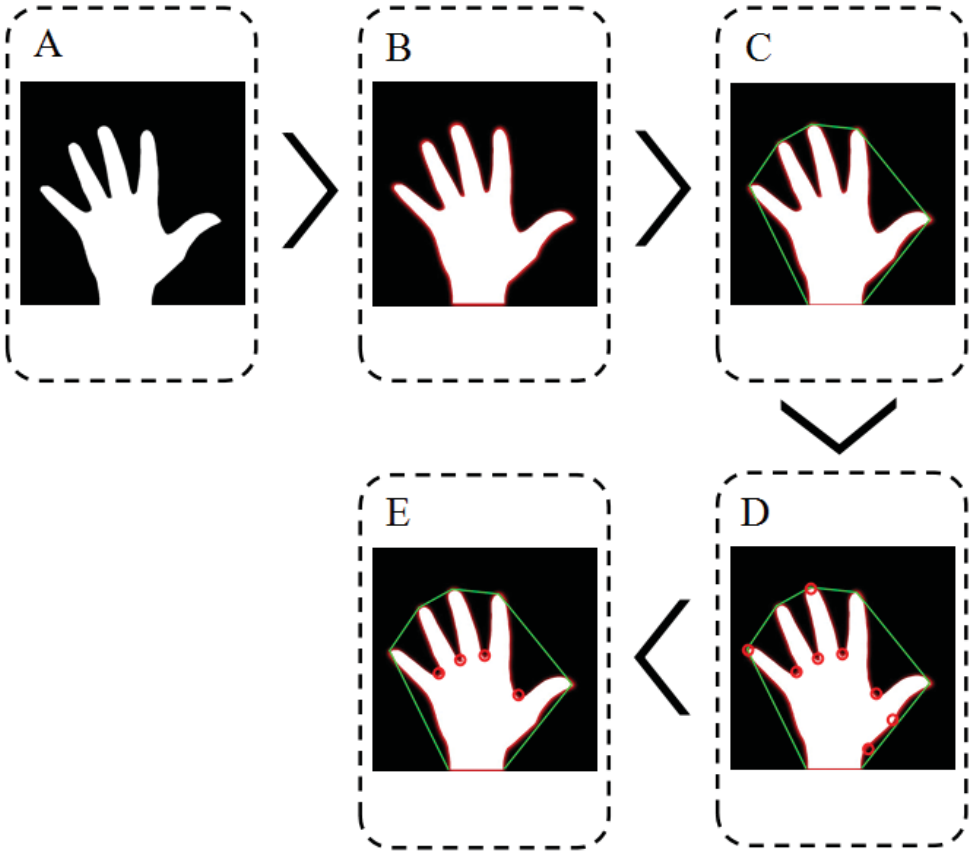

Figure 4.31: The process of recognizing gesture from the binary image

Next, we introduce the process and using functions in each step:

Step 1. Find contours.

We use an OpenCV function findContours()[83]to find contours of the given binary image:

```
void findContours(InputOutputArray image,
OutputArrayOfArrays contours,    OutputArray hierarchy,    int mode,
intmethod, Point offset=Point())
```

It is built based on the border following algorithm [83]. The parameter "mode" defines the contour retrieval mode. In this case, we used CV_RETR_EXTERNAL to retrieve the most outer contours. The parameter "method" represents the approximation method, and here we use CV_CHAIN_APPROX_NONE which

means the results can contain all points of the contour. The result is shown in step B of Figure 4.31, the red curve represents the contour.

Step 2. Find convext hull of a point set

As complex contour of an object is unsuitable for pattern detection, we need to retrieval the convex hull of the hand. We use R. L. Graham and F. F. Yao [73]'s algorithm here. The key function is OpenCV function convexHull() [91]:

```
void convexHull(InputArray points,                    OutputArray hull,
boolclockwise=false, bool returnPoints=true )
```

It can find the convex hull as shown in step C of Figure 4.31. The green curve is the convex hull of the hand.

The other important algorithm in this step is Ramer-Douglas-Peucker algorithm. It can use a simpler linear curve to replace the original curve. We use this algorithm to simplify the convex hull we get. The key function is called approxPolyDP() [91]:

```
void approxPolyDP(InputArray curve,     OutputArrayapproxCurve,
double epsilon, bool closed)
```

Step 3. Find points furthest away from the convexity defects.

To recognize the gesture, we need to figure out how many fingers are in the image. The number of fingers can be calculated based on the convexity defects of the contours. Figure4.32 shows the convexity defects of the contour line around the hand. There are eight area (A, B, C, D, E, F, G, H) represent convexity defects. What we need is the furthest point of each convexity defects.

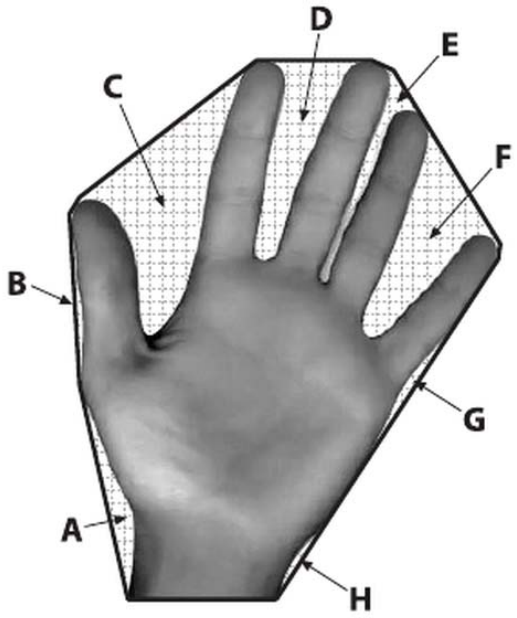

Figure 4.32: An example of convexity defects of the contours [92]

The key function in this step is OpenCV function called convexityDefects() [91]:

void convexityDefects(InputArray contour, inputArrayconvexhull, OutputArrayconvexityDefects)

The output array contains the information of the furthest point of each convexity defect. The red circles in step D of Figure 4.31 represents the points.

Step 4. Discard irrelevant points.

The rule to determine whether the points from last step is relevant are shown in Figure 4.33. The points A and B are the start points of convex hull. All start points can be get from function convexityDefects(). If the length is less than $0.4l_{bb}$ and angle is bigger than 80degrees. The point C is irrelevant. Through this step, we can get the result as shown in the step E of Figure 4.31.

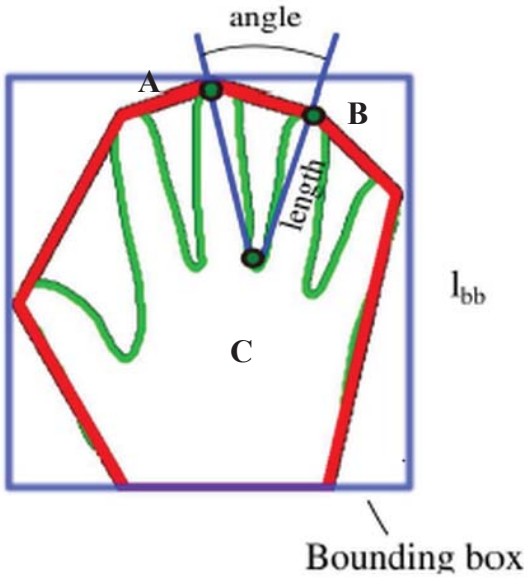

Figure 4.33: The rule to determine whether the points are relevant

4.4.1.5 Show information of the gesture

Figure 4.34 shows the vision of gesture recognition. The fingertips are indicated with green circles. And the center of the palm is represented as a small red rectangle. The number of defects are shown in the bottom of the image.

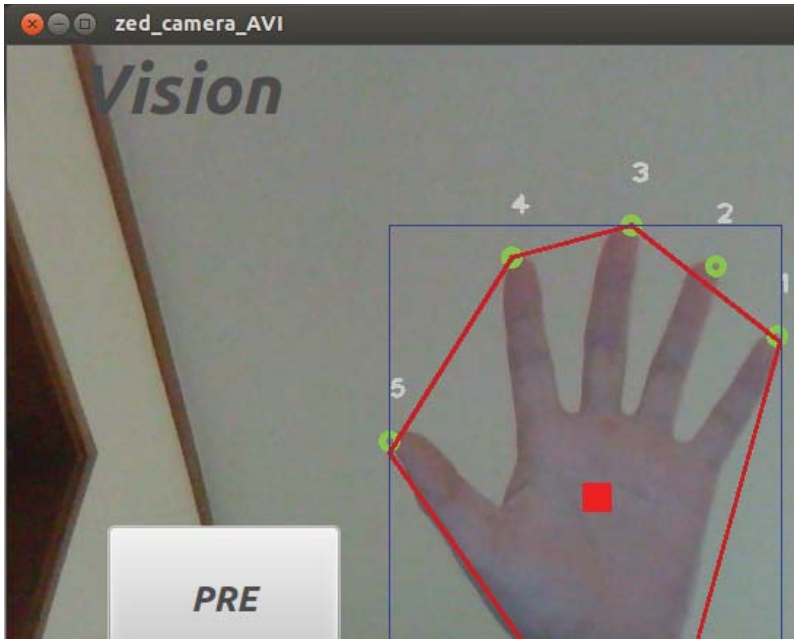

Figure 4.34: The vision of gesture recognition

4.4.2 Using ZigBee Module to Control Device

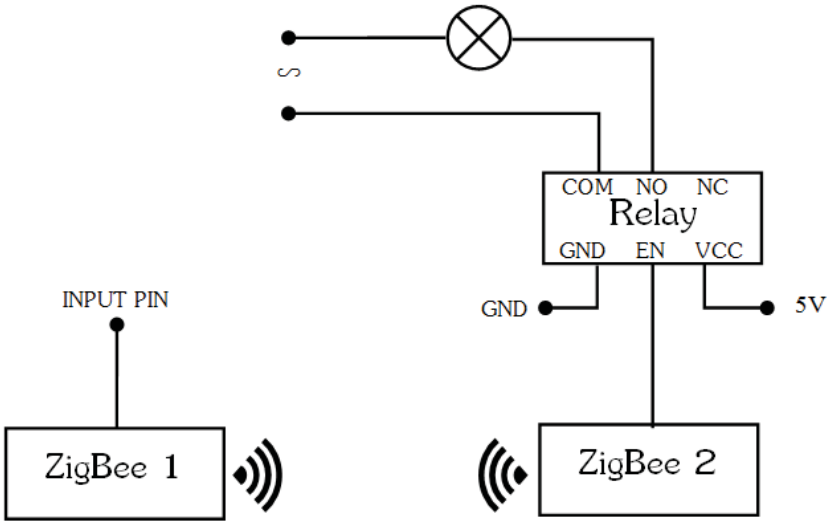

Figure 4.35: The wiring diagram of ZigBee communication

Take lamp as an example, Figure 4.35 shows how to wire up the lamp. The input pin of ZigBee module 1 is connected with the development board through GPIO ports. ZigBee module 1 is used to receive signal from development board and send signal to ZigBee module 2. ZigBee module 2 is used to receive signal from ZigBee module 1 and send signal to relay. The relay is connected with the lamp. When the input pin of ZigBee 1 receive a high signal, the lamp will be switched on, otherwise, the lamp will be switched off.

The development board send the control signal through GPIO interface. We need to configure the GPIO ports to communicate with the ZigBee module first. There are three steps to configure a GPIO pin to output a high signal: 1. open the pin; 2. set direction as out; 3. set value as high. Assume we use GPIO 133, the function to open the lamp is :

```
open_lamp(void)
{
   //open gpio 133
   int fd=open("/sys/class/gpio/export",O_WRONLY);
   if(fd<0)
   {
      cout<<"can't open export"<<endl;
      return fd;
   }
   write(fd,"133",3);

   //set direction as out
   fd=open("/sys/class/gpio/gpio133/direction",O_WRONLY);
   if(fd<0)
   {
      cout<<"can't open direction"<<endl;
      return -1;
   }
   write(fd,"out",3);

   //set value as 1
   fd =open("/sys/class/gpio/gpio133/value",O_WRONLY);
   if(fd<0)
   {
      cout<<"can't openc value"<<endl;
      return -1;
   }
   write(fd,"1",1);

   return 0;
}
```

4.4.3 Using Voice Recognition to Recognize Voice Commands

A voice recognition module based on DTW algorithm is applied for voice recognition. Figure 4.36 shows the process of voice recognition. There are two phases: DTW training phase and DTW testing phase. Through pre-processing stage, the input signal which is analog signal is converted into digital signal. Mel-frequency cepstrum coefficients are used to extract features. In the training phase, the sampling templates are generated based on the training data. In the testing phase, the similarity between the sampling templates and the input templates.

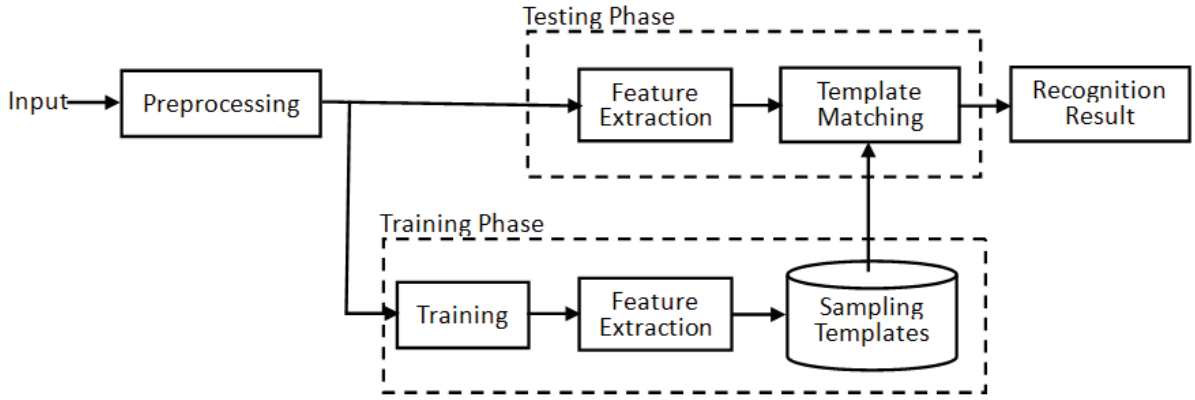

Figure 4.36: The process of voice recognition

Before connect the voice recognition module with the development board, we need to record the commands first according to the manual [93]. The commands we use are shown in Table 4.3. If the commands are recognized, the module send the corresponding string through UART interface.

Table 4.3: Voice commands

Commands	Output string
Open	0x11
Off	0x12
Lamp	0x13
Fan	0x14

The voice recognition module can communicate with the development board through the serial ports. We need to configure the serial port of development board: open the port, set baud rate, set data bits and set other parameters.

When the module recognize a voice command, it send the corresponding string to development board. We need to read the string and then switch on/off the relative device.

4.4.4 Using Gesture and Voice Recognition to Control Devices

A prototype is built as a simple example to control appliance through the Internet of things using gesture and voice control, shown in Figure 4.37. It can be divided into two stations:

1. Control station. It is responsible for real-time image processing, gesture recognition, voice recognition and sending signal to the appliance station. There are mainly five components in control station: USB camera, LCD display, ZigBee module, voice recognition module and development board. The ZigBee module can receive signal from development board through GPIO interface and send signal to the other ZigBee module in appliance station. Voice recognition module is connected to development board through UART interface and capable to recognize 15 kinds of voice commands. In this case, we only use four different voice commands.

2. Appliance station. It includes two appliances: one ZigBee module and three relays. The ZigBee module can receive signal from the other one and send corresponding signal to enable or disable relays. There are three relays: one is connected to lamp's switch and the other two control fan's two speed respectively.

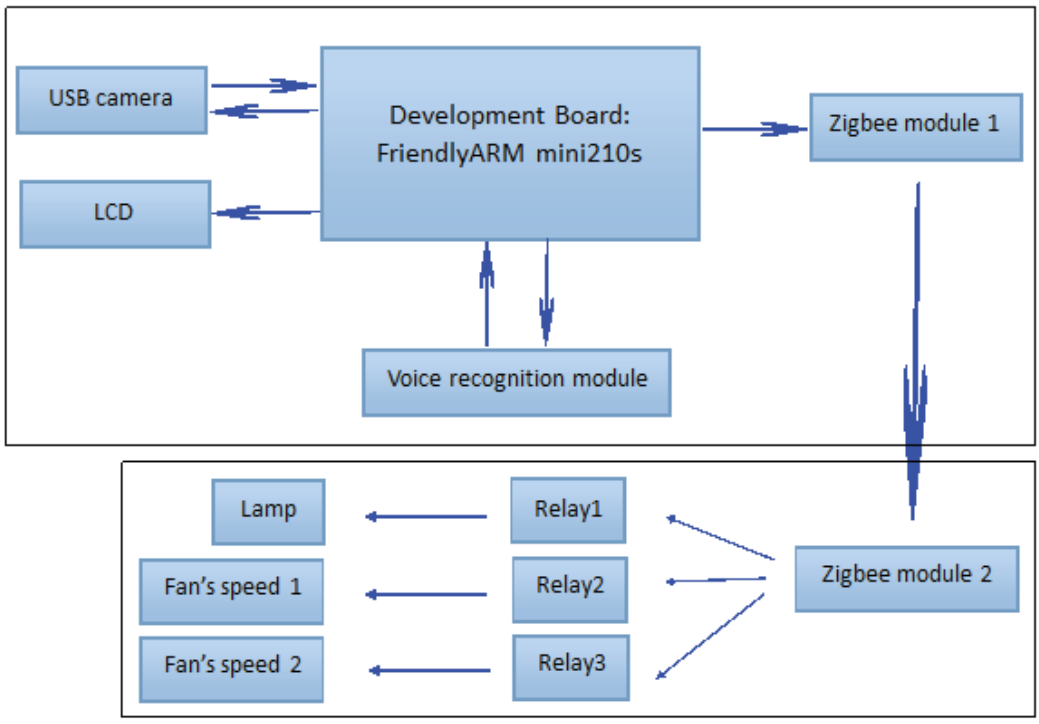

Figure 4.37: The structure of the prototype

4.4.5 User Interface

There are three pages of interface which can be easily switched through buttons: NEXT, PRE and Vision. We use the center of palm to represent the hand, as shown in Figure 4.34. Every button on the interface can be switched on/off if the user hand stays on it more than three seconds.

The vision page is to display the real-time image of gesture recognition as shown in Figure 4.34. In this page, basic information of gesture is shown, including if hand true/false, number of fingers and the coordinates of hand's center.

The fan panel and the lamp panel are shown in Figure 4.38. Fan and lamp can also be controlled using voice commands. When receives voice command "Fan", fan will be turned on and the speed is low. When receives voice command "Lamp", lamp will be lighted. Command "Open"/Off" is to turn on/off the fan/lamp only if the current showing page is fan/lamp panel.

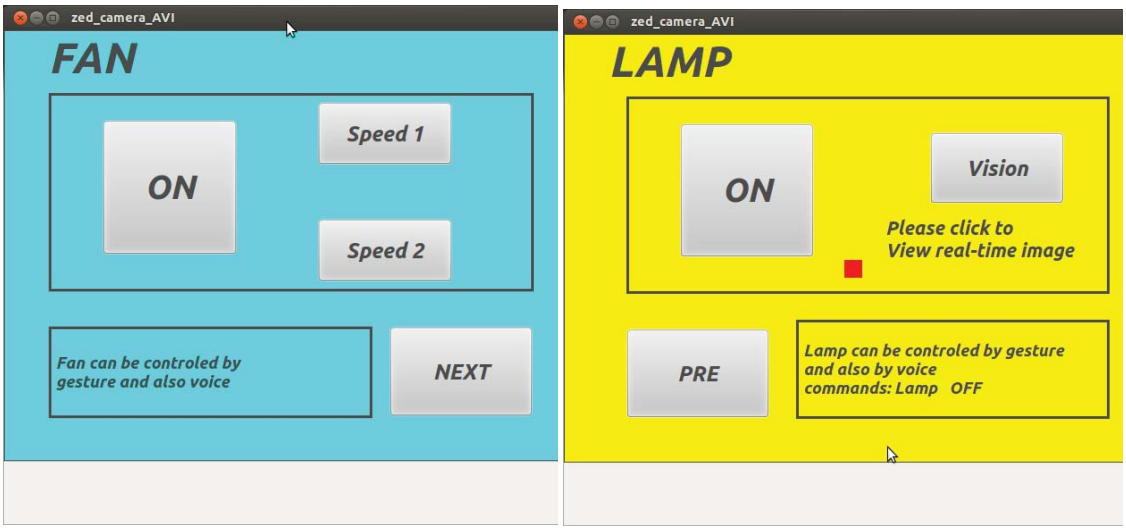

Figure 4.38: The Fan and Lamp panel

CHAPTER 5 TESTSANDRESULTS

5.1 Introduction

With the prototype we built, we were able to control the appliances through gesture and voice control.

After hand color sampling, user's palm can be recognized and the center of the palm is represented as a red rectangle on the screen. The user can control the rectangle by moving hand. Every button on the screen can be toggled when the rectangle keeps staying on it for 3 seconds. With the gesture control, the user can successfully control the lamp on and off, set the fan on different speed and shift between different appliance control panels.

In separate appliance control panel, voice control can also be activated when a correct command is captured by the microphone. There are four voice commands: Lamp, Fan, Open and Off. The command "Lamp" can switch on the lamp while the command "Fan" can turn the fan on. And the command "Open"/"Off" can switch on/off the appliance only if the current showing panel is the corresponding panel.

With the early test, hand recognition can be perfectly performed if the environment has no color similar to the color of the hand. The user can switch on/off the appliances with gesture control after some exercises. In terms of voice control, the commands can be recognized clearly if there is not much background noise. In order to analyze the reliability and accuracy of the prototype, the following tests were carried out.

5.2 Test for Gesture Control

Test one focuses on the accuracy and efficiency of the gesture control. In preparation, we found in suitable environments (background has no similar color to the color of the user hand), the user hand can be recognized with very high accuracy. In this test, we only test in suitable environments. As indoor/outdoor is

not an important factor on the performance of gesture control, the test is executed in indoor environment. Five people are asked to complete a series of operations using gesture control. We analyze the factors of accuracy and efficiency based on the time they use to complete the operation.

Each tester needs to complete seven operations as Figure 5.1 shows: First, after the system gets the sample color of the tester's hand, the fan panel is shown automatically. The tester needs to use gesture control to turn on the fan. When the fan is on, the tester switches it to speed 2 immediately. Next step is to turn it off. Then click "NEXT" button to switch to the lamp panel. The tester needs to turn on the lamp. Then turn it off. In the end, the user needs to activate the "PRE" button to go back to the fan panel.

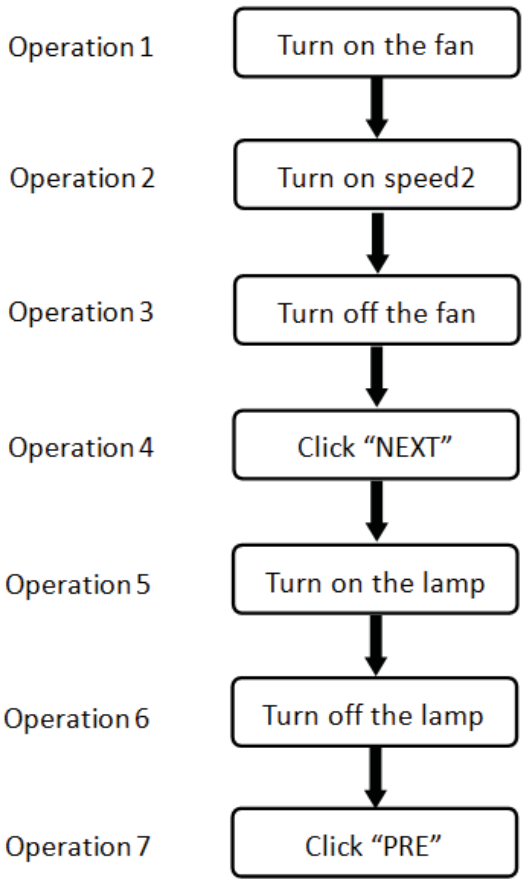

Figure 5.1: The process of test 1

Time1 is from the point that the fan panel appears to the point that the lamp panel appears. Time2 is from the point that the lamp panel appears to the point that the fan panel appears. As every button takes at least three seconds to be toggled, the minimum time should be $time1_{min}$ = 12 seconds, $time2_{min}$ = 9 seconds. Table 5.1 shows the data we get.

Table 5.1: The results of test 1

Tester	Time1(s)	Time2(s)	Total(s)
Tester1	15	12	27
Tester 2	14	10	24
Tester 3	15	10	25
Tester 4	16	11	27
Tester 5	20	15	35

The results of test 1 show the system is quite stable in suitable environment. All values are reasonable and have no big differences, showing the system can recognize gesture correctly and quickly.

Tester 2 uses the least of time in operation because tester 2 has lots of practice before the test. To the contrast, tester 5 uses the most of time due to the unfamiliarity to the system. When the tester moves hand too far (more than 900mm according to the test), the system cannot capture the hand because the object is too small. But if the tester moves hand too near (less than 200mm according to the test) the camera, the system cannot recognize it either. This is because when the hand is too near, the color of the hand captured by the camera is very different from the sampling color. So the training is necessary for the user to be familiar with the system to control movement of hand within the effective range. During the test, when the hand is not in the effective range, the system shows no hand captured, then the testers are asked to adjust the distance by themselves, so they can all finally get the commands done with different completion time.

5.3 Test for Voice Control

Test 2 aims at the accuracy of the voice control. During the test, 5 testers are asked to use all four voice commands to complete the operations in different environments. If the system failed to recognize the command, the tester needs to repeat the command until success. We will analyze the accuracy of the voice control based on the number of times that each tester needs to speak for each command. The testers need to complete the operations shown in Figure 5.2.

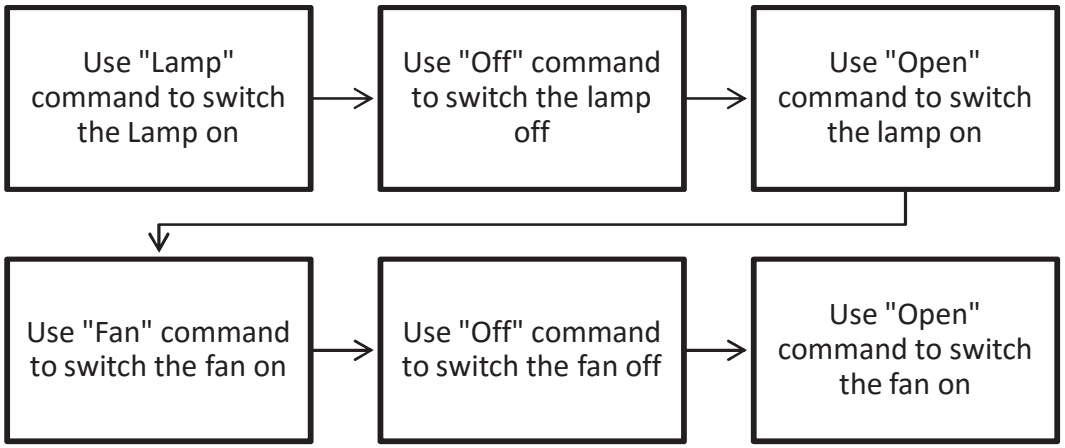

Figure 5.2: The process of test 2

All of the six operations are repeated by each tester in two different environments: indoor and outdoor. The indoor environment can be regarded as an environment that has no outstanding background noise. And the outdoor environment is a place close to a busy street with frequent traffic noise and wind noise. The results of the tests are shown in Table 5.2.

Table 5.2: The results of test 2

Environment	Command	Times that tester speaks the command				
		Tester1 (times)	Tester2 (times)	Tester3 (times)	Tester4 (times)	Tester5 (times)
Indoor	1	1	1	2	1	1
	2	1	1	1	1	2
	3	1	2	1	2	1
	4	1	1	1	1	1
	5	2	1	1	2	1
	6	1	1	1	1	1

Outdoor	1	3	2	2	4	4
	2	2	3	4	5	2
	3	3	5	2	4	1
	4	4	3	4	3	5
	5	3	5	1	5	4
	6	3	4	3	3	3

As Table 5.2 shows, all the testers can complete the test in the indoor environment using relatively less repeating times. Most of the commands are successfully completed by speaking once. While in the outdoor environment, the testers usually need to speak several times. The average times that testers speak for each command are shown in Figure 5.3.

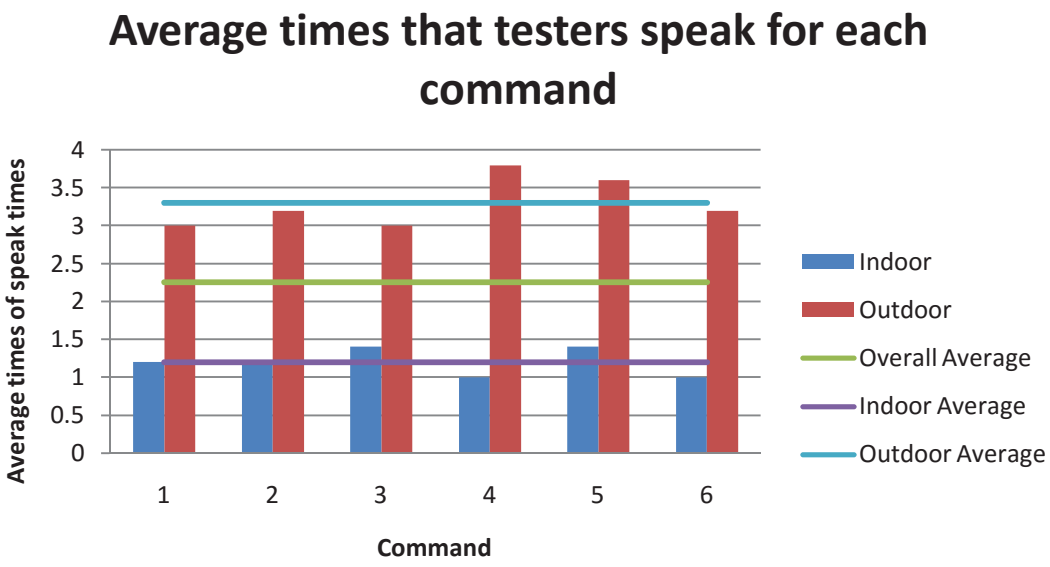

Figure 5.3: Average times that testers speak for each command

The average times of the outdoor test are three times higher than the indoor test. When the test is carried out indoor without much background noise, most of the operations can be accomplished by only speaking the command once. The system has very high accuracy and efficiency. However, in the outdoor environment, the performance of the system is very poor. Gesture control is not affected by indoor/outdoor environment, so it could be a good compensated method in the outdoor environment.

5.4 Summary of the Tests

The two tests prove that the system is effective in suitable environments. There are four points need to be considered to apply this system:

1. Gesture control can perform correctly and efficiently only if the background has no color similar to the color of the hand.
2. Voice control can be very efficient in indoor environments without much background noise. But in noisy environments, the performance is very poor and unstable.
3. The users need a little training for using gesture control. They need to be familiar with the control system.
4. Similar voice commands can be very confusing for the system. Like in this prototype, we choose four very different commands, so the accuracy is very high.

CHAPTER 6 CONCLUSION AND FUTURE DIRECTION

6.1 Conclusion

The goal of this project was to develop an embedded system to control the Internet of things using gesture and voice recognition. The border following algorithm, convex hull algorithm and Ramer-Douglas-Peucker algorithm were applied to accomplish gesture recognition while DTW algorithm was used to perform voice recognition. Two ZigBee modules were employed as wireless connection modules. An user interface was developed with the ability to add more control panels for different situations.

At the beginning of the thesis an overview of the Internet of things was given. The history and landscape of IoT were introduced. The components of a typical IoT system were shown. As voice and gesture control were used as the input methods, the basic concepts, technologies and applications of using voice and gesture recognition in IoT were discussed.

Next we reviewed the related literature focusing on the existing technologies and algorithms, including the basic concepts, principles and features. Five related works were critically reviewed and analyzed specifically. The algorithms, advantages and disadvantages of each related work were discussed. Skin-based algorithm is effective but unstable under different lighting conditions in contrast with Haar algorithm which could reach high accuracy under different conditions. 3D gesture recognition is more accurate than 2D algorithms while non-vision based approach is more simple and efficient but less flexible than vision based approach.

Following the literature review, the system design, integration and implementation were presented. The algorithms for gesture control and voice control were introduced. There are three key algorithms for gesture control: (i) The border following algorithm which is to get the contours of the hand; (ii) The

convex hull algorithm which is for extracting the convex hull of the contours; (iii) The Ramer-Douglas-Peucker algorithm to simply the convex hull. The algorithm for voice control is DTW algorithm.

The platform for the embedded system involved porting the embedded Linux operating system to the development board and building the cross compilation environment.

Following the set-up of the Linux-based embedded platform, software modules were designed for the system. It recognized the gesture from real-time frame and voice commands through the voice recognition module. An user interface was developed and ZigBee modules were integrated to perform wireless communication. The devices were controlled by relays connected to the ZigBee module.

The hardware devices in this thesis were wireless connection modules, a camera, a voice recognition module, relays and appliance devices. Two ZigBee modules were used as wireless connection modules. A web camera was employed as input device while a voice recognition module was used to recognize certain voice commands. A lamp and a desk fan were selected and wired through relays to the ZigBee module as the appliance devices to control.

The system could be expanded to control more appliances by adding specific appliance stations for each additional appliance. The main components in appliance station were the relays and the ZigBee module. The ZigBee modules and the relays were chosen depending on the requirements of the appliances and made flexible for multiple appliances.

Finally, system tests were performed and the results were evaluated. Two tests were undertaken to analyze the performance of gesture control and voice control. Test for gesture control was under ideal environment (the background has no color similar to the color of the hand). Five testers completed a series of operations. According to the time they used, the factors of accuracy and efficiency were analyzed. Test for voice control was in two different environments: indoor

environment (where there is no outstanding noise) and outdoor environment (where there is frequent traffic noise and wind noise). The results showed the significant change of accuracy under the two kinds of environments. The system performed much more accurately and efficiently under the indoor environment.

6.2 Contributions

This thesis has five main contributions:

1. Extract hand from background in real time frame with high accuracy.
2. Recognize gesture by counting fingertips and collect other information like the position of palm and the area of gesture.
3. Recognize different voice commands and allow user to adjust commands.
4. Develop friendly user interface which is open to add more control panels for new appliances.
5. Enable communication between the embedded board and appliances through ZigBee modules.

6.3 Future Direction

For future work for this project, several directions can be considered.

First, more advanced voice recognition method can be used to support more complicate voice commands. The voice recognition system may have the ability of self-evolution to improve the accuracy based on self-evaluation. When the system receives a voice command, it may send audio output to confirm the command. The microphone also can be upgraded as input voice from relatively far distance cannot be received very well.

In this thesis, we proposed an approach to recognize gesture by counting the numbers of fingers. Based on the numbers of fingers, it is capable of recognizing six kinds of gestures (fist, gesture with one/two/three/four fingers, open palm). In

future work, other control commands based on different gestures can be added, like fist which can mean to close all devices.

Another future direction is to add more control method like mind-control and facial recognition. Facial recognition can help to identify users and provide individual services. Mind-control is a new approach and showing promising development.

More sensors can be used in the system to adjust different appliances. For example, temperature sensor can be used to perform indoor temperature control.

In future research, the system can be connected to the Internet. For the development board used in the thesis, it can be connected to the Internet using network gateway through the integrated RJ45 interface. Furthermore, the ZigBee modules can be replaced by recently standardized ZigBee IP devices which can support both ZigBee radio frequencies and IPv6 routing protocol. By connecting to the Internet and adding cloud computing, data center and mobile applications, the system would become an IoT of anything, anywhere, anytime.

REFERENCES

[1] J. Höller *et al.*,*From Machine-to-Machine to the Internet of Things: Introduction to a New Age of Intelligence*, Waltham, MA: Academic Press, 2014.

[2] Firdaus *et al.*, "ZigBee and WiFi Network Interface on Wireless Sensor Networks", in *Proc. 2014 Electrical Engineering and Informatics Int. Conf.*, Makassar, Indonesia, 2014

[3] *Intel Xeon Processor E7 V3 Family* [Online]. Available: http://ark.intel.com/products/family/78585/Intel-Xeon-Processor-E7-v3-Family#@Server

[4] San Diego. (2016, Feb 12). *Qualcomm Announces Next Era of Wearables with New Snapdragon Wear Platform* [Online]. Available: https://www.qualcomm.com/news/releases/2016/02/11/qualcomm-announces-next-era-wearables-new-snapdragon-wear-platform

[5] List of Intel CPU Microarchitectures [Online]. Available: https://en.wikipedia.org/wiki/List_of_Intel_CPU_microarchitectures

[6] J. Anderson and L. Rainie. (2014, May 14). *The Internet of Things Will Thrive by 2025* [Online]. Available: http://www.pewinternet.org/2014/05/14/internet-of-things/

[7] S. Hiremath, "Wearable Internet of Things: Concept, Architectural Components and Promises for Person-centered Healthcare", in *Proc. 2014 Wireless Mobile Communication and Healthcare Int. Conf.*, Athens, Greece, 2014.

[8] H. Rubine, "The Automatic Recognition of Gestures," Ph.D. dissertation, CMU.,PA,1992.

[9]M. Gorman.(2013, July 22). *Leap Motion Controller Review* [Online].Available: http://www.engadget.com/2013/07/22/leap-motion-controller-review/

[10] J. Melià-Seguí, "RFID EPC-Gen2 for Postal Applications: A Security and Privacy Survey", in *Proc. 2010 RFID-Technology and Applications Int. Conf.*, Guangzhou, China, 2010

[11] G. Lawton, "Machine-to-Machine Technology Gears Up for Growth", in *IEEE Computer Society*, vol. 37, issue 9, New York, NY:IEEE, 2004, pp. 12-15.

[12] *Sometimes, Less Power is More* [Online]. Available: https://www.qualcomm.com/products/wifi-platforms

[13] *Brain Power* [Online]. Available: http://research.ibm.com/cognitive-computing/neurosynaptic-chips.shtml

[14] *Bluetooth/ Bluetooth Low Energy* [Online]. Available: http://www.st.com/web/en/catalog/sense_power/FM1968/CL1976/SC1898?sc=bluetoothlowenergy

[15] *Smart Cities Seoul: a Case Study*[Online].Available: https://www.itu.int/dms_pub/itu-t/oth/0b/15/T0B150000153301PDFE.pdf

[16] P. Clarke (2012, Dec 4). *Who Has the Lowest Power MCU?* [Online]. Available: http://www.st.com/web/en/catalog/sense_power/FM1968/CL1976/SC1898?sc=bluetoothlowenergy

[17] *The Internet of Things* [Online]. Available: http://raymondjames.com/pointofview/article.aspx?a=2023

[18] J. McQuivey (2014, April 16).*Your Voice Will Control the Future* [Online]. Available: http://blogs.forrester.com/james_mcquivey/14-04-16-your_voice_will_control_the_future

[19] G. Beavis, (2014, Dec 10). *Android Wear: Everything You Need to Know* [Online]. Available: http://www.techradar.com/news/portable-devices/google-android-wear-what-you-need-to-know-1235025

[20] A. Pham(2009, Jun 1). *E3: Microsoft Shows Off Gesture Control Technology for Xbox 360*[Online]. Available: http://latimesblogs.latimes.com/technology/2009/06/microsofte3.html

[21] N. M. Richardson(2013). *One Giant Leap for Mankind* [Online]. Available: http://www.inc.com/30under30/nicole-marie-richardson/leap-motion-david-holz-michael-buckwald-2013.html.

[22] P. Kumar *et al.,* "Hand Data Glove: A Wearable Real-Time Device for Human-Computer Interaction, "in *International Journal of Advanced Science and Technology,* vol. 43: SERSC, 2012.

[23] Y. Liu *et al.*(2013), *SoundSense: 3D Gesture Sensing Ultrasound on Mobile Devices* [Online].Available: http://mrorz.github.io/files/soundsense.pdf

[24] *MB1010 LV-MaxSonar* [Online]. Available: http://www.maxbotix.com/Ultrasonic_Sensors/MB1010.htm

[25] *Infrared Gesture Sensing*, Silicon Laboratories, Inc., Austin, TX, 1996.

[26] *Overlay Considerations for the Si114x Sensor* [Online]. Available: https://www.silabs.com/Support%20Documents/TechnicalDocs/AN523.pdf

[27] Y. Kim and C. Moon, "Non-Contact Gesture Recognition Using the Electric Field Disturbance for Smart Device Application", in *International Journal of Multimedia and Ubiquitous Engineering*, Vol 9, Issue 2, 2014.

[28] E. Kuronen, *Epic Sensors in Electrocardiogram Measurement* [Online]. Available: http://www.theseus.fi/bitstream/handle/10024/67543/Kuronen_Esa.pdf?sequence=1

[29] B. Kellogg *et al.,* "Bringing Gesture Recognition to All Devices," in *Networked Systems Design & Implementation,* Berkeley, CA, 2014, pp. 303-316

[30] J. Wu *et al.*, "Gesture Recognition with a 3-D Accelerometer", in *Proc. 6th on Ubiquitous Intelligence and Computing Int. Conf.*, Brisbane, Australia, 2009, pp. 25-38.

[31] *ADXL330-Analog Devices* [Online]. Available: http://www.analog.com/en/products/mems/mems-accelerometers/adxl330.html

[32] A. H¨ofer *et al.*, "Gyroscope-based Conducting Gesture Recognition", in *Proc.2009 New Interfaces for Musical Expression Int. Conf.*, New York, NY, 2009.

[33] H. Ketabdar *et al.*, "MagiTact: interaction with mobile devices based on compass (magnetic) sensor", in *Proc.15th Intelligent User Interfaces Int. Conf.*, Hong Kong, China, 2010, pp.413-414.

[34] R. Matthews *et al*.(2007, Sep 1). *Nonintrusive, Wearable Bioelectrodes for Monitoring the Heart and Brain* [Online]. Available: http://www.sensorsmag.com/specialty-markets/medical-devices/nonintrusive-wearable-bioelectrodes-monitoring-heart-and-bra-1412

[35] G. R. S. Murthy and R. S. Jadon, "A Review of Vision Based Hand Gestures Recognition", in *International Journal of Information Technology and Knowledge Management*, vol. 2, no. 2, Switzerland: Inderscience, 2009, pp. 405-410.

[36] J. P. Wachs *et al.*, "Vision-based Hand-gesture Applications", in *Communications of the ACM*, vol. 54, issue 2, New York, NK: ACM, 2011.

[37] Y. Wu and T. S. Huang, "Non-stationary color tracking for vision-based human computer interaction," in *IEEE Transaction on Neural Networks*, vol. 13, no. 4, New York, NY: IEEE, 2002, pp. 948–960.

[38] S. K. Kang *et al.*, "Color Based Hand and Finger Detection Technology for User Interaction", in *Proc. 2008 Convergence and Hybrid Information Technology Conf.*, Busan, South Korea, 2008, pp. 229-236.

[39] Q. Chen *et al.*, "A comparative study of Fourier descriptors and Hu's seven moments for image recognition", in *Proc. IEEE Canadian Electrical and Computer Engineering Conf.*, vol. 1, Ontario, Canada, 2004, pp. 103–106.

[40] J. Marnik, "Hand Shape Recognition for Human-Computer Interaction", in *Man-Machine Interactions*, vol. 59, Berlin, Germany: Springer, 2009,pp. 95-102.

[41] A. Mittal *et al.* (2011).*Hand detection using multiple proposals* [Online].Available: http://www.robots.ox.ac.uk/~vgg/publications/2011/Mittal11/mittal11.pdf

[42] S. Malik *et al.*, "Hand tracking for interactive pattern-based augmented reality," in*Proc.2002 Mixed and Augmented Reality Int. Symp.*, Darmstadt, Germany, 2002, pp 117-126

[43] D. G. Lowe, "Object Recognition from Local Scale-Invariant Features", in *Proc.7thComputer Vision Int. Conf.*, vol. 2, Kerkyra, Greece, 1999, pp. 1150–1157.

[44] X. Han *et al.*, "Real-time scene recognition on embedded system with SIFT Keypoints and a New Descriptor", in *Proc. 2013 IEEE Mechatronics and Automation Int. Conf.*, Kagawa, Japan, 2013, pp. 1317-1324.

[45] H. Bay *et al.*, "SURF: Speeded Up Robust Features", in *Proc. 9thEuropean Conf. on Computer Vision*, 2006, pp. 404-417.

[46] H. Zhang, "Fast image matching based-on improved SURF algorithm", in *Proc. 2011 Electronics, Communications and Control (ICECC) Int. Conf.*, Ningbo, China, 2011, pp. 1460-1463.

[47] E. Oyallon and J. Rabin. (2013).*An Analysis and Implementation of the SURF Method, and its Comparison to SIFT*[Online]. Available: http://www.ipol.im/pub/pre/69/.

[48] D. Comaniciu *et al.,* "Real-time Tracking of Non-rigid Objects Using Mean Shift", in *Proc. 2000 IEEE Computer Vision and Pattern Recognition Conf.*, vol. 2,Hilton Head Island, SC, 2000, pp. 142–149.

[49] K. Nummiaro *et al.*, "An Adaptive Color-based Particle Filter", in *Image Vision Computing*, vol. 21, no. 1 : Elsevier B.V., 2003, pp. 99–110.

[50] Chetan. S and Dr. M. Z. Kurian, "An Effective Algorithm for Tracker Operation for Gesture Recognition System", *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering*, vol. 2, issue 7, India :S&S, 2013.

[51] Q. Chen, "Real-Time Vision-Based Hand Tracking and Gesture Recognition, "Ph.D. Dissertation, University of Ottawa, Ottawa, Ont., Canada, 2008.

[52] C. F. Shan *et al.*, "Real-time Hand Tracking Using a Mean Shift Embedded Particle Filter", in *Pattern Recognition*, vol. 40, issue 7,London, UK: Elsevier B.V., 2007, pp. 1958-1970.

[53] P. Viola and M. Jones, "Robust Real-time Object Detection", in *International Journal of Computer Vision*, Germany: Springer, 2001, pp. 1-3.

[54] Q. Chen and N. D. Georganas, "Hand Gesture Recognition Using Haar-Like Features and a Stochastic Context-Free Grammar", in *IEEE Transactions On Instrumentation And Measurement,* vol.57, no.8, New York, NY :IEEE, 2008, pp. 1562-1571.

[55] C. Tomasi *et al.*, "3D tracking = classification + interpolation", *in Proc. 9th IEEE Computer Vision Int. Conf.*, vol. 2, Nice, France, 2003, pp. 1441-1448.

[56] H. Zhou and T. Huang, "Okapi-Chamfer Matching for Articulate Object Recognition", in *Proc. 10thComputer Vision Int. Conf.*, Beijing, China, 2005, pp. 1026-1033.

[57] G. N. Meenakshi and P. K. Ghosh, "Automatic Gender Classification Using the Mel Frequency Cepstrum of Neutral and Whispered Speech: a Comparative Study", in *Proc. Twenty First National Conference on Communications*, Mumbai, India, 2015, pp. 1-6.

[58] S. D. Dhingra *et al.*, "Isolated Speech Recognition Using MFCC and DTW", in *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering*, vol. 2, issue 8, India: S&S, 2013.

[59] A. Jain *et al.* (2010).*Real Time Speech Recognition Engine* [Online]. Available: http://people.ece.cornell.edu/land/courses/ece5760/FinalProjects/f2010/np276_ksp55_aj355/np276_ksp55_aj355/

[60] S. Salvador and P. Chan, "FastDTW: Toward Accurate Dynamic Time Warping in Linear Time Space", in *Intelligent Data Analysis*, vol. 11, no. 5: IOS Press, 2007, pp. 561-580.

[61] A. Sannino, "Analyzing Discontinuous Speech in EU Conversations: A Methodological Proposal, Journal of Pragmatics", in *Journal of Pragmatics*, vol. 38: Elsevier B.V., 2006, pp. 543-566.

[62] J. Kirriemuir, "Speech recognition technologies." Retrieved December 5 (2003): 2005.

[63] S. R. Thite *et al.*, "Speech Recognition Using DTW", in *Advances in Computational Sciences and Technology,* vol. 5, no.1, Pune, India: RIP, 2012, pp. 1077-1083.

[64] L. E. Baum and T. Petrie, "Statistical Inference for Probabilistic Functions of Finite State Markov Chains", in *the Annals of Mathematical Statistics,* vol. 37, no. 6: IMS, 1966, pp. 1554 – 1563.

[65] N. Najkar *et al.*, "A Novel Approach to HMM-based Speech Recognition Systems Using Particle Swam Optimization", in the Mathematical and Computer Modelling, vol. 52, Issue 11-12: Elsevier B. V., 2010, pp. 1910-1920.

[66] S. A. Zahorian *et al.*, Vowel Classification for Computer based Visual Feedback for Speech Training for the Hearing Impaired," in *7th Spoken Language Processing Int. Conf.*, Denver, CO, 2002.

[67] J. J. Choondal and C. Sharavanabhavan, "Design and Implementation of a Natural User Interface Using Hand Gesture Recognition Method", in *International Journal of Innovative Technology and Exploring Engineering*, vol. 2, issue 4: BEI-ESP, 2013.

[68] K. Madhuri and L. P. Kumar,"Cursor Movements Controlled By Real Time Hand Gestures," in *International Journal of Science and Research*, vol.2, issue 2, 2013.

[69] K. Rimkus *et al.*"3D Human Hand Motion Recognition System", *in 6th Human System Interaction Int. Conf.*, Sopot, Poland, 2013, pp. 180-183.

[70] R. Neßelrath *et al.*"A Gesture System for Context-Sensitive Interaction with Smart Homes", *Ambient Assisted Living*, Berlin, Germany: Springer Berlin Heidelberg, 2011, pp. 209-219.

[71] S. Suzuki and K. Abe. "Topological Structural Analysis of Digitized Binary Images by Border Following", in *Graphical Models, Computer Vision Graphics and Image Processing*, London, UK: Elsevier B.V., 1985, vol. 30, no. 1, pp. 32-46.

[72] I. T. Young *et al.*(1995).*Fundamentals of Image Processing* [Online]. Available:

http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/TUDELFT/FIP2_3.pdf.

[73] R. L. Graham and F. F. Yao. "Finding the Convex Hull of a Simple Polygon", in *Journal of Algorithms*, London, UK: Elsevier B.V., 1983, vol. 4, Issue 4, pp. 303-412.

[74] *Ramer-Douglas-Peucher Algorithm* [Online]. Available: http://en.wikipedia.org/wiki/Ramer%E2%80%93Douglas%E2%80%93Peucker_algorithm.

[75] L. Thiele and E. Wandeler, *Performance Analysis of Distributed Embedded Systems*[Online].Available:http://www.tik.ee.ethz.ch/file/b86cabc530475fa279c895e8989daf99/TW05.pdf

[76] M. Fang, *Embedded Linux Primer* [Online]. Available: https://mfcoding.wordpress.com/linux/embedded-linux-primer/

[77] *About Us* [Online]. Available: http://www.qt.io/About-Us/

[78] *Mini210s | S5PV210 ARM Cortex-A8 Board* [Online]. Available: http://www.friendlyarm.net/products/mini210s

[79] *10moons V804 Camera* [Online].Available:http://www.helloipad.com/computer-accessories-android-tablets-pc/192091602-dropshipping-discount-10moons-v804-webcam-with-cx202ar-built-in-noise-canceling-microphone-led-lights-pc-camera-webcams.html

[80] *Arduino Voice Recognition Module* [Online]. Available: http://www.geeetech.com/wiki/index.php/Arduino_Voice_Recognition_Module.

[81] *XBee Buying Guide* [Online]. Available: https://www.sparkfun.com/pages/xbee_guide.

[82] *Pololu Basic SPDT Relay Carrier with 5VDC Relay* [Online]. Available: https://www.pololu.com/product/2480.

[83] *Micasa 6" Desk Fan* [Online]. Available: http://www.harveynorman.co.nz/home-appliances/heating-and-cooling/fans/micasa-6-desk-fan.html

[84] *Mi Casa Caitlin Table Lam*p [Online]. Available: http://www.thewarehouse.co.nz/red/catalog/product/Mi-Casa-Caitlin-Table-Lamp-Grey?SKU=1861499

[85] QT-*About Us* [Online]. Available: http://www.qt.io/About-Us/

[86] R. Sandberg et al. "Design and Implementation or the Sun Network Filesystem," in *Innovations in Internetworking*, Norwood, MA: Artech House, 1988, pp. 379-390.

[87] *Video4Linux* [Online]. Available: https://en.wikipedia.org/wiki/Video4Linux

[88] *Functions Open* [Online]. Available: http://pubs.opengroup.org/onlinepubs/009695399/functions/open.html

[89] *Operations on Arrays* [Online]. Available: http://docs.opencv.org/modules/core/doc/operations_on_arrays.html

[90] *Image Filtering* [Online]. Available: http://docs.opencv.org/modules/imgproc/doc/filtering.html

[91] *Structural Analysis and Shape Descriptors* [Online]. Available: http://docs.opencv.org/modules/imgproc/doc/structural_analysis_and_shape_descriptors.html

[92] *Contours* [Online]. Available: http://compvis.readthedocs.org/en/latest/contours.html

[93] *Voice Recognition Module* [Online]. Available: http://www.geeetech.com/Documents/User%20Manual.pdf

[94] P. Senin (2008), *Dynamic Time Warping Algorithm Review* [Online]. Available: http://www2.hawaii.edu/~senin/assets/papers/DTW-review2008draft.pdf

[95] *Defining Connectivity* [Online]. Available: http://www.imageprocessingplace.com/downloads_V3/root_downloads/tutorials/contour_tracing_Abeer_George_Ghuneim/connectivity.html