ST. XAVIER'S COLLEGE

(Affiliated to Tribhuvan University)

Maitighar, Kathmandu

Final Year Project Report

on

"Gesture Recognition in Virtual Reality and Its Application in Education"

[CSC-404]

A Final Year Project Report submitted in the partial fulfillment of the requirement for the degree of Bachelor of science in Computer Science and Information Technology awarded by Tribhuvan University

Under the Supervision of

Mr. Vishnu Kumar Rana

HoD, Department of Computer Science

Submitted By:

Arun Sanjel (T.U. Exam Roll No. 2677/070)

Submitted To:

ST. XAVIER'S COLLEGE

Department of Computer Science

Maitighar, Kathmandu,

Nepal

August 2017

ST. XAVIER'S COLLEGE
MAITIGHAR, KATHMANDU, NEPAL
Post Box :7437
Contact: 4221365,4244636
Email: ktm@sxc.edu.np

सेन्ट जेभियर्स कलेज
माईतीघर, काठमाडौं, नेपाल
पो.ब.नं. : ७४३७
फोन : ४२२१३६५,४२४४६३६
ईमेल : ktm@sxc.edu.np

CERTIFICATE OF APPROVAL

The undersigned certify that they have read and recommended to the Department of Computer Science for acceptance, a project proposal entitled "**Gesture Recognition in Virtual Reality and Its Application in Education**" submitted by **Arun Sanjel (TU Roll No.: 2677/ 070)** for the partial fulfillment of the requirement for the degree of Bachelor of Science in Computer Science and Information Technology awarded by Tribhuvan University.

…………………………..
Vishnu Kumar Rana
Lecturer/ Supervisor
Department of Computer Science
St. Xavier's College

…………………………..
External Examiner
Tribhuvan University

……………………………..
Vishnu Kumar Rana
Head of the Department
Department of Computer Science
St. Xavier's College

ACKNOWLEDGEMENT

With immense pleasure, we express our deepest sense of gratitude and sincere thanks to our highly respected and esteemed guide **Mr. Vishnu Kumar Rana –** Head of Department and our supervisor for his confidence bestowed on us and for entrusting our project through his valuable guidance. Therefore, we would like to express our innermost thanks to him for providing us with all the crucial advices, guidelines and resources for the accomplishment of this project.

We are also grateful to the entire Computer Science Department of St. Xavier's College for housing us a seemly environment where we could work with this project. We were pleased to be under the commands of the department to help us from all possible ways. We would also take this opportunity to express our gratitude to **Er. Rajan Karmacharya** for his continuous encouragement and support throughout the completion of this project.

We would also like to express our heartfelt gratitude to **Mr. Jitendra Manandhar, Mr. Bal Krishna Subedi, Er. Anil Shah, Er. Saugat Sigdel, Mr. Ganesh Yogi, Mr. Nitin Malla, Mr. Sansar Dewan, Er. Sanjay Kumar Yadav, Mr. Ganesh Dhami** and **Mr. Ramesh Shahi** for their constant support and guidance. Furthermore, we are also appreciative towards all our colleagues, seniors and relatives who had directly or indirectly been a part of this case study.

Arun Sanjel (TU ROLL NO.: 2677/070)

ABSTRACT

Virtual reality is an artificial environment that is created with software and presented to the user in such a way that the user suspends belief and accepts it as a real environment. Virtual Reality in its simplest form is a simple 3-D image that can be explored interactively using a computer with graphics capabilities. Computer Vision refers to the science which enables a computer to understand images and videos and extract meaningful information from them. This technology has made many of the current trends like Robot Navigation, Self-Driving Cars, etc. possible. Immersive learning refers to the process of learning by putting the learner in an interactive environment. The environment is the main part in learning processes where immersive learning is used.

The ways with which one can interact with the environment in a Virtual Reality application is fairly limited. There are some Bluetooth based controllers which can be used for interacting with the environment but most of these devices feel far from natural. Another option is to buy expensive hardware which do feel natural which makes access to such devices difficult. Thus, the combination of Computer Vision and Virtual Reality is slowly gaining ground in the form of Augmented Reality (AR). Augmented Reality merges the physical world with the virtual world in order to create a more immersive experience for the user.

This project aims to combine some of the currently emerging trends in Computer Science; Computer Vision and Virtual Reality, in order to create a higher level of interactivity. This heightened interactivity enables new immersive experiences in a number of applications that make use of Virtual Reality. It makes use of a smartphone mounted on a cheap Head Mounted Display(HMD) and the same phone's camera. The user can interact with the virtual environment by using their bare hands, without the need for any external hardware like data gloves.

Keywords: Virtual Reality, Computer Vision, Image Processing, Immersive Learning, Gesture Recognition, Stereoscopic Images

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

VR	Virtual Reality
AR	Augmented Reality
MR	Mixed Reality
RGB	Red, Green, Blue
HSV	Hue, Saturation, Value
HSL	Hue, Saturation, Lightness
HIS	Hue, Saturation, Intensity
YCbCr	Luminance; Chroma: Blue, Chroma: Red
YIQ	Luminance, In-phase, Quadrature
YUV	Luminance and Two Chroma Components
3D	Three Dimensional
2D	Two Dimensional
NLP	Natural Language Processing
VLE	Virtual Learning Environment
ICT	Information and Communications Technology
IDE	Integrated Development Environment
SDK	Software Development Kit
API	Application Programming Interface

CHAPTER 1: INTRODUCTION

1.1 Background

This project aims at providing a cost-effective and natural method of interaction in Virtual Reality(VR) as well as Augmented Reality(AR) applications.

Gesture Recognition is an interaction technique popularly used by humans to interact with computers. It is a technology of computer science with the goal of interpreting human gestures or human motion via mathematical algorithms. Gesture recognition begins from facial, voice, eye tracking, lip movement, ear etc. Also, the identification and recognition of posture, walking style, and human behaviors is also the part of gesture recognition techniques [1]. Gesture Recognition has been implemented in a number of contexts the most familiar one being on touch based mobile devices. Though voice recognition is gaining ground in recent years, other forms of Gesture Recognition are not accessible so easily. Particularly, applications of human body based gesture recognition techniques are hard to find.

However, with the rise of VR and AR applications, new devices capable of taking data from the real world and merging them with digital data are becoming more common. One of the first attempts to do this was the Google Glass which has now been discontinued. Another product to have gained popularity in the world of AR is Microsoft HoloLens which has yet to be released to non-developers. The research in the field of VR has also increased. 364 papers have been published between 2009-2011 in the IEEE and ISMAR VR conferences [2].

The problem with all the current VR techniques which feel natural is that they are either hard to get access to or are too expensive for the general public. Even good mobile VR headsets which are the most cost-effective form of experiencing Virtual Reality have a large price tag. The Google Cardboard, however, is the most accessible VR headset on the market. On top of that, there a number of other headsets available which are based on the Cardboard's original design.

This project aims to introduce a cost-effective way of interaction with virtual reality through the use of a Google Cardboard based VR headset.

1.2 Problem Statement

The field of virtual reality is gaining momentum today but this field still lacks interaction techniques that feel natural and are affordable. This is especially true in Mobile Virtual Reality applications. Motion/object detection systems like the Kinect are expensive as well as complex for integration with immersive Virtual Reality. VR interaction hardware and technologies are not only expensive but there is an absence of adequate amount of applications that can fully them. This is because of they are quite hard to gain access to. Some VR applications make use of motion and movement detection but the APIs and documentations related to them are limited.

Virtual Reality is being used today in many domains, one of which is the field of learning. Current education system and learning process is very monotonous and lacks interaction. Students cannot gain all the knowledge that they need from a single book or text sheet. This leaves room for improvement in the imagination and brainstorming capabilities of students. Thus, learning is a field that can benefit tremendously from the use of Virtual Reality but is limited by the lack of natural interaction techniques.

1.3 Project Objectives

- To design a Computer Vision system which can perform hand and fingertip tracking.
- To enable gesture recognition with the help of the hand and fingertip tracking data.

- To create a natural interaction technique in Virtual Reality and Augmented Reality systems.
- To help developers to easily integrate hand and gesture detection capabilities into their applications.
- To demonstrate the potential of Virtual Reality based learning aided by the natural feel of hand motion and gesture recognition.
- To provide a cheaper way to enjoy a more natural feeling Virtual Reality experience.
- To make the project documentation.

1.4 Scope of the Project

The use of hand tracking and gesture recognition in the world of virtual reality can prove to be a new way of experiencing virtual reality. It enables users to interact with the virtual world as if they were interacting with the real world. This combination can aid in many fields that make use of Virtual Reality, including but not limited to learning. For example, being able to rotate a 3D model of a molecule by using bare hands and even zooming in, in order to view the atoms inside it can prove as an effective learning technique in a chemistry class. This is just one of the huge array of possible applications of this technology. Also, this is an era where there is not enough time for developers to code every single thing so they rely on APIs created by third parties. Up until now, there have not been many APIs for hand tracking and gesture recognition through the use of computer vision. This problem is also addressed by this project.

1.5 Features of the Project

This project focuses of creating an API using which users can track hand movement and gestures through the use of a camera. The main features of this project are:

- Hand Tracking API: Developers can use the classes and methods from this API to create their own applications which leverage hand tracking.

- Demo Application: It provides a demonstration on how the API can be used to create an application to make education more interactive.

1.6 Requirement Analysis

Requirement analysis is one of the initial tasks performed in system engineering. It is usually comprised of studying the existing system and system requirement specification.

1.6.1 Existing System

1.6.1.1 Microsoft HoloLens

Figure 1 Microsoft Hololens [3]

Microsoft HoloLens is a VR headset which consists of transparent lenses on which graphics, data, or information can be displayed for an Augmented Reality experience. It is capable of doing all of it's processing by itself and can perform advanced operations like spatial mapping and object recognition. It can also track the user's body parts and enables the use of hand gestures to control the virtual and augmented environment [4].

1.6.1.2 Pebbles Interfaces

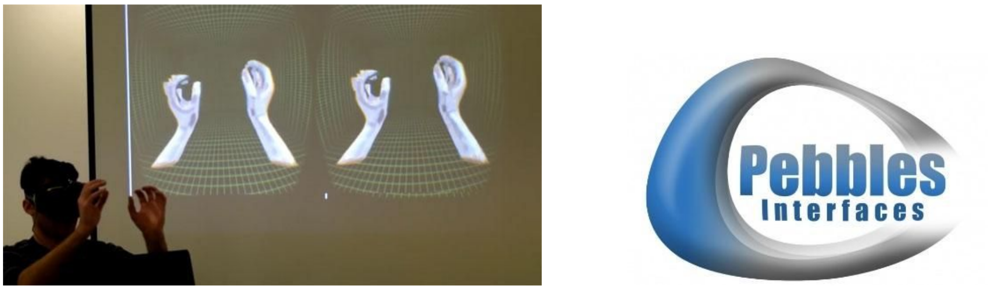


Figure 2 Pebbles Interfaces [5] [6]

Pebbles Interfaces is a company based on Israel which has spent five years developing hardware and software systems in order to detect and track hand movement. They aim to take virtual reality to the next level by improving the information that can be extracted by optic sensors. Pebbles has recently been acquired by Facebook's Oculus [7].

1.6.1.3 Kinect

Figure 3 Kinect [8]

The Kinect is a motion sensing device which was intended as an add-on for Microsoft's Xbox 360. The Kinect features a depth camera which can track objects and movement in three dimensions. The Kinect removes the need for a controller and allows the player to engage with the game directly by using their body's motions. It can recognize and differentiate between different players through face and voice recognition features [9].

1.6.2 Data Collection

It is very important to collect data from lots of sources in order to make a project possible. Many researchers have documented their work on virtual reality, hand/object detection and gesture recognition systems in various journals. These papers were studied in order to gain knowledge on the techniques currently being used for these purposes as well as to gain a clear understanding of the performance complexity of the project. The different algorithms and techniques discussed in the papers were compared and studied in order to decide the best approach.

1.6.3 System Requirement Specification

1.6.3.1 Functional Requirements

- Hand Segmentation: The system should be able to segment the user's hand from the frames captured by the camera.
- Fingertip Detection: It should be able to locate the fingertips from the isolated hand.
- Gesture Recognition: It should be able to recognize gestures that the fingertips make.

1.6.3.2 Hardware Requirements

The minimum hardware requirements to run the project are as follows:

1.6.3.2.1 For Immersive Virtual Reality

- Virtual Reality Headset:
    - Google Cardboard based headset
    - Compartment that exposes smartphone camera

- Android Smartphone:
    - 1GB RAM
    - ARM Processor
    - Back Camera
    - Gyro Sensor
    - Accelerometer

1.6.3.2.2 For Hand Tracking in Other Applications

- Webcam
- A computer system with:
    - 2GB RAM or higher
    - I3 Processor

1.6.3.3 Software Requirements

The minimum software requirements of this system are:

- Operating System: 64-Bit Windows Operating System, Android KitKat or higher
- For Development using the API: Visual Studio or other C++ IDEs
- For creating Virtual Reality Applications: Unity, Unreal Engine or other software capable of creating Virtual Reality applications

1.7 Feasibility Analysis

A feasibility study is done to determine whether the project is technically, economically and operationally feasible. This study helps us know if the project is viable and will also help govern different situations where problems may arise during implementation. The feasibility analysis carried out in this project are elaborated below.

1.7.1 Economic Feasibility

This project aims to create a cost-effective solution for interaction. The current tools and technologies in the market like the Kinect, Oculus Rift and the HoloLens cost hundreds and thousands of dollars. This project utilizes a Google Cardboard based VR headset which costs as low as $5. Most people have a smartphone around so this is not much of an issue. A smartphone which meets the mentioned requirements can be bought in the range of $100 to $200. The Android One series of devices give the most value for money [10].

1.7.2 Technical Feasibility

The technical feasibility study assesses the details of how the project delivers a product or service. This study defines if a project can be feasible in terms of technical requirements or not. After a detailed feasibility study, following conclusions were obtained:

- There exist libraries for Image Processing which can be used to perform complex image processing tasks
- Cameras are available in almost all devices
- Upgrading of the project to add new API features and classes is easy
- The hardware and software requirements are not too high

1.7.3 Operational Feasibility

Any project that can be implemented in real world remains beneficial. If it cannot be implemented, it exists only in theoretical approach. This project is feasible operationally. The interaction is natural using the fingertips of the user. This will require the user to learn to use some gestures but most people are already used to hand gestures through the use of smartphones.

1.7.4 Legal Feasibility

Implementation of this project does not violate any rules or standards defined by the government of Nepal or any principles internationally. The concepts and techniques taken from other sources are properly cited. The concept is based solely on the analysis of team members, thus does not violate any copyright act.

1.7.5 Schedule Feasibility

In case of schedule feasibility, major time is consumed by the research and analysis part. Once an operational program has been developed, the project reaches its completion with the exception of updating the API as newer and faster algorithms become available. For this analysis, a time schedule was created prior to the initiation of project and a deadline on which it must be completed. The project has a minimal schedule for completion and can be completed within the defined period.

CHAPTER 2: LITERATURE REVIEW

The overall goal of this project is to add a form of natural interaction to Virtual Reality applications. This goal can be realized by combining Virtual Reality with the science of Computer Vision. This project needs to deal with technologies relating to both Virtual Reality and Computer Vision. On top of that, because the end product of this project contains an app which shows the use of our project in education, it is also necessary to study the effectiveness of Virtual Reality in education. This requires a thorough research in all the three fields, each of which have their own challenges.

2.1 Virtual Reality

2.1.1 Stereoscopic Images/Video

Virtual Reality requires the use of two different angles of the same image in each frame of the output video, where each image simulates the image seen by each eye. This type of image is called a stereoscopic image. According to a study was done by Paul Hands and Jenny C.A. Read, where they compare different techniques to produce stereoscopic images in order to give a sense of depth to the video, a stereoscopic 3D image created by simply shifting a 2D image is not immersive [11]. So, it is very important to render two different angles of our 3D scene in order to give a truly immersive experience. This can prove to be computationally expensive [12].

2.1.2 Relativistic Virtual World

Virtual Environments have been a staple of science fiction and engineering innovation for decades, but are becoming an enabling technology in fields ranging from medicine to entertainment. Yet there is still much work to be done in the area of virtual representation. This is particularly true as applications become increasingly unconventional. Insights from a wide range of scientific fields might be able to help out conceptually by tying together human behavior and virtual worlds using concepts from relativistic approaches [13].

To understand why high-dimensional, relativistic spaces could be useful in application to virtual environments, we must consider where the research on control of virtual environments is headed. In the early years of virtual environments, control of virtual worlds was limited to simple interaction devices such as keyboards, computer mice, and joysticks. These devices provided sparse, discrete information to a virtual environment [14]. At the same time, the individual who controls a virtual environment ultimately uses their brain and body to realize action in this synthetic world. Early in the development of virtual environments, it was not considered necessary to couple action in the virtual world and physical control [15].

2.2 Computer Vision

Computer Vision is a core part of this project. It is important to ensure that this project balances the accuracy of hand and gesture recognition with the computational complexity.

2.2.1 Hand Recognition and Segmentation

There are many approaches to segmenting out a hand from a given image or frame. For colored image processing; several color spaces are commonly applied [16]. Any image acquired by a digital camera(s) or video is decoded by RGB color model [17] which is the most common color space, where the components of each color pixel are defined by red, green, and blue in an orthogonal Cartesian space [16] according to the Tristimulus theory [17] where the three bands form the human visual system [16]. Color models that depend on human color perception are various. HSV color model is one of these models, which has been used widely used. Almost every approach of hand detection involves some form of skin detection using a range of color values that represent the skin [18] [19] [20]. Numerous color spaces with different characteristics are used for skin color segmentation, and most of them share same properties [21]. The main four commonly used are RGB, HSV (and similar HSI and HSL), YCbCr (with similar YIQ and YUV), and CIE Lab (and CIE Luv) [22]. RGB color space defines the primary colors in three channels, red, green, and blue. The high correlation between these channels leads to a luminance sensitivity color space, and provides RGB color space with different properties for any lighting condition changes. For this reason, other color spaces have been used for skin detection purposes [16].

The next step in hand segmentation is thresholding. Thresholding is the process of separating the hand detected in the image from the background. Thresholding works by setting a threshold value and extracting all the pixels that fall inside this threshold value. In this case, the threshold value used is the color space corresponding to skin color. Thresholding produces a binary image where all pixels that fall within the threshold value are white and all other pixels are black. The figure below shows the results of hand segmentation in the RGB color space by using three different threshold values, 70 and 20. This example was demonstrated by Dhawan et. al [23]

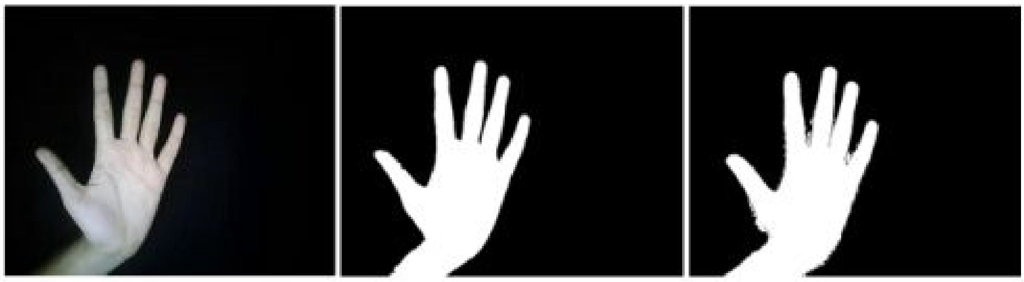


Figure 4 Segmentation in RGB Space using different threshold values (left: original image; center: threshold 70; right: threshold 20) [23]

Some other researchers, however, have turned to using neural networks for detecting the presence of hands [24] [25]. The part where we see a difference is in the hand posture recognition systems. Some researchers have chosen to rely solely on the use of image processing to find the defining characteristics of hand postures in the video stream [18]

[19]. The other approach is to use neural networks and train them to recognize various postures as well as gestures [26] [24] [27].

Verma suggested a method to extract features of hands from the captured image such as fingertips, edges and vectors and then use them for 2D modeling. He used something called the harris corner detector in order to detect and extract fingertip information [28]. Nguyen made the use of gray scale morphology and geometric calculations in order to relocate fingertip locations. He did this by using a learning based model on a 640x480 pixel size frame [29].

2.2.2 Gesture Recognition

Touch based gesture recognition systems have been around for long but creating a computer vision based gesture recognition system is more complex compared to a touch based system. "The human hand is a complex deformable object and gesture itself has many characteristics, such as diversities, ambiguities, temporal and spatial differences and human vision itself is an ill-posed problem" [30]. Choi has pointed out that these algorithms can have large processing times which increase the computational complexity. He says "the most important issue in field of the gesture recognition is the simplification of algorithm and the reduction of processing time". He used morphological operation to implement his system using the center points extracted from primitive elements by morphological shape decomposition [31].

Most gesture recognition techniques make use of a state machine or grammar in order to determine the gestures once the features of the hand have been determined. Verma used c-mean fuzzy clustering based finite state machines (FSM) in order to recognize hand gestures. Formula for centroid calculation of fuzzy c-means clusters is that centroid is the mean of all points weighted by their degree of belonging to the cluster center [28]. Trivino made a more descriptive system which can convert human gesture positions into a language like description by the use of fuzzy logic. He related it to Natural

Language Processing (NLP). He used sensors and took only few positions in sitting and standing into consideration [32].

Another way of determining gestures is to simply track the motion of the hand [33] and extract features of the path and the hand to recognize the gesture. The real challenge is in accurately recognizing hand features. The detected features are processed in order to find the gesture.

Nolker focuses on large number of 3D hand postures in her system called GREFIT. She used finger tips in hands as natural determinant of hand posture to reconstruct the image. She suggests marked and colored fingertips and making a histogram as well as using different reference prototypes or images. This method makes use of 192x144 size gray scale images for processing [34].

Some researchers tried their hands on neural networks in order to recognize hand gestures. One such example is Stergiopoulou, who used unsupervised Self-Growing and Self-Organizing Neural Gas network. This system works on 31 different gestures. He made the following assumptions:

1. The hand position is always vertical
2. The user uses only the right hand

Thus, this system does not work for left handed people. The raised fingers detection in the hand is done by using the fingertip neuron, which is followed by another neuron chain as shown in the figure. Calculating the center of the palm is important. This is done by using the gravity method by using neurons that lie in the palm area and distance from the fingertips to palm center is calculated. One of the main problems in gesture recognition is that only raised fingers are detected in most algorithms. This takes away our ability to detect more complex gestures. Stergiopoulou then applied a likely hood classification to get the current gesture from a pool of predefined gestures [35].

Figure 5 SGONG Network Working (a) start with two points (b) growing stage with 45 neurons (c) output with 83 neurons (d) hand gesture (e) only raised fingers would be counted [35]

2.3 Education

The use of this gesture detection technology with Virtual Reality will prove to be an effective tool in various domains including learning as found in a research that focused on how direct manipulation is better than passive viewing for learning anatomy [36]. The interest of students to latest information and communication technology (ICT) applications and tools, particularly online and network games, multimedia, social networking, chat, and Internet surfing, has been recently persevering. Students are more curious and attentive towards the virtual world. So, shifting from traditional classes to Virtual Learning Environment (VLE) in education is great idea. Creating a virtual class where student can interact with every object and manipulate them will help them understand and learn interactive way and making them less monotonous. The understanding and brain storming capacity of student also increases with higher curiosity in educational topics. Using simulator, Open world helps them to deal with real life problem in virtual environment which appears to be as real as possible [37].

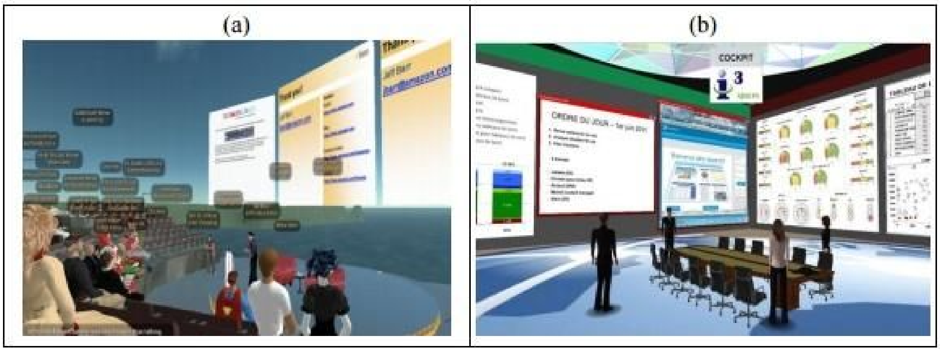


Figure 6 A screenshot of an example learning session in the point of view of one of the participants (a) – Using Sl/TSL; (b) – Using OWL [37]

2.3.1 ICT Based Education

By the introduction of ICT based education, the geographical difficulties are removed. Not only it provides platform for varieties of student but also it increases the creative and innovation skills of the students. ICT gives a common platform for many students to put through their ideas, opinions and suggestions which increases the exposure of the students [38]. ICT based education have been practiced and implemented in major developed nation. According to recent study, country like China is trying to fully implement ICT based education by 2020. The developed nation like USA, ICT based education is well practiced and utilized by student where students to computer ratio is 3:1 and also all the schools are connected to internet. This infrastructure helps in the developed nation to easily implement ICT based project [39].

Technologies allow students to work more productively than in the past, but the teacher's role in technology- rich classrooms is more demanding than ever. ICT has the potential to transform the nature of education (improving teachers' design work, enhancing the roles of students and teachers in the learning process and helping to create a collaborative learning environment, etc.) Yes, ICT has a great potential in extending of education system but in developing nation with less infrastructure and resources, it is far from reaping its benefits. Not only the infrastructure is the problem in implementing this methodology but also various other factor of nation like insufficient funds, proper planning, political factors and more so ever corruption in the developing nation is hinder its development [40].

3.2. E-Learning and M-Learning

Electronic Learning (E-learning) and Mobile Learning (M-learning) are the two vital buzz terms in modern education particularly in Education Enhanced Technology and Technologies Supported Learning. E-learning is defined as the “instructional content or learning experience delivered or enabled by electronic technologies” whereas, M-learning is defined simply as learning via mobile devices such as cell phones, smart phones, palmtops, and handheld computers. There are many similarities between the two technologies as both are modern learning tools. These two technologies are equally used by the educational institutions and sometimes students can access the same content as per their wish and convenience. There are many advantages and disadvantages in both the systems. Most of the technical limitations are fixed by the growth in the technology. The success of these technologies now lies with the content management [41].

3.3. Blended Learning

Figure 7 Blended Learning

The implementation of virtual learning has great advantages with some pitfalls too. The term Blended learning is introduced here. Since e-learning environments present some disadvantages such as inhibiting the socialization process of individuals resulting in lack of face-to-face communication; a new environment has surfaced. This new environment combines the e-learning and the classical learning environments.

Major benefit of blended learning is presence of teacher and face to face communication with every entity. In addition, blended learning has the potential to change students' experiences and outcomes through learning. The above discussion has identified the major benefit of applying blended instruction i.e. to overcome the shortcomings of online instruction and exploit various instructional process and delivery strategies in order to increase learners' satisfaction as well as boosting the learning outcomes [42].

CHAPTER 3: SYSTEM DEVELOPMENT

3.1 Project Management Strategy and Tools

A project is a group of activities designed to produce a unique product, service or result. Project management is the application of knowledge, skills and techniques to execute projects effectively and efficiently. Project management is the discipline of planning, organizing, and controlling resources to achieve specific goals. Project management has been necessary and important in this project. The constraints for this project, as for most projects, have been time, cost and quality. Project management is necessary to complete the project under these constraints and utilize the resources properly [43]. The project management process can be categorized into five groups – initiating, planning, executing, monitoring and controlling and closing [44].

The overall development of the system was carefully analyzed under the proper guidance from the supervisor and team members.

3.1.1 Project Work Flow and schedule

- Team Size: 2
- Total project duration: 3 Months

3.1.2 Project Team

Team Resource	**Role**
Mr. Vishnu Kumar Rana	Supervisor
Bikash Paneru	Developer/Designer
Arun Sanjel	Developer/Designer

Table 1: Team Resource and Roles

Table 1 shows the various people associated with the project. The Project Supervisor is responsible for guiding the team members and observing their working procedures and techniques. The project members are responsible for carrying out the jobs as directed by the supervisor.

3.1.3 Responsibilities

Each project aims to solve a problem or improve an already existing solution. A project which cannot solve the intended problem is of no use. It is also important to properly plan the project and stick to that plan in order to effectively complete the project. All of the team members should thoroughly understand their responsibilities.

Responsibilities of The Supervisor

- Project Scheduling
- Schedule Tracking
- Sharing Information
- Documentation

Responsibilities of The Team Members

- Preliminary Research
- Background Study
- Feasibility Study
- Design and Analysis
- Development and Testing
- Implementation and System Evolution
- Draft report writing and submission
- Final report writing and submission

3.1.4 Development Model

Each development organization has its own working process that it evolves for how it gets software development done. Although that might sound chaotic, in practice there is pretty broad consensus on what the constituent steps are that make up the entire process. So, it becomes a matter of arranging those steps, of tweaking the details, and especially of settling on the relative emphasis and level of detail in these steps. The arrangements that companies arrive at are seldom entirely innovative. Instead, they fall into a few standard patterns [45]. The Software Development Models are the various processes or methodologies that can be selected for the development of the project depending on the project's aims and goals.

This project best because it consists of a number of components which rely on each other. This also requires that a component is coded only after the component which it depends on has been coded. Thus, the incremental model best suits this project.

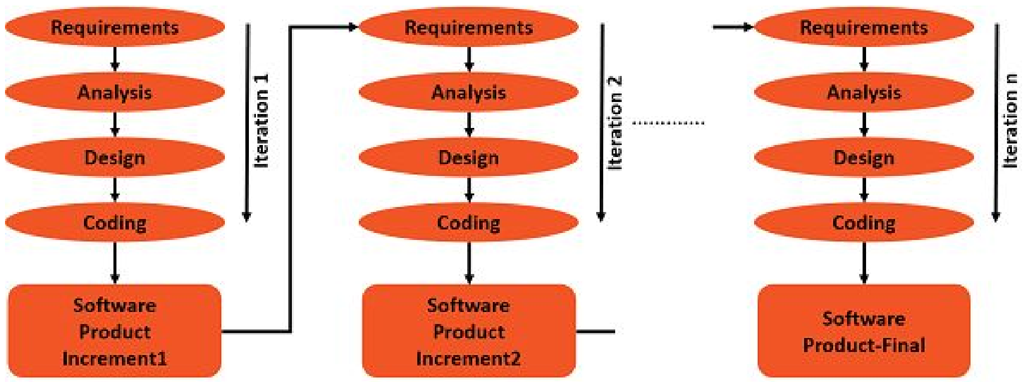


Figure 8 The Iterative Model [46]

In incremental model, the whole requirement is divided into various builds. Multiple development cycles take place here, making the life cycle a "multi-waterfall" cycle. Cycles are divided up into smaller, more easily managed modules. Each module passes through the requirements, design, implementation and testing phases. A working version of software is produced during the first module, so you have working software early on during the software life cycle. Each subsequent release of the module adds function to the previous release. The process continues till the complete system is achieved [47]. First, a partial implementation of a total system is constructed so that it will be in a deliverable state. Increased functionality is added in later stages. Defects, if any, from the prior delivery are fixed and the working product is delivered. The process is repeated until the entire product development is completed. The repetitions of these processes are called iterations. At the end of every iteration, a product increment is delivered [48].

3.1.5 System Development Tools

Some of the tools used for the completion of this project are:

3.1.5.1 Microsoft Visual Studio 2017

Microsoft Visual Studio is an Integrated Development Environment (IDE) which provides various tools and services for creating a wide array of applications including windows applications, web applications, node.js applications, mobile apps, etc [49]. It also includes various other developmental tools and can be used to aid development with a lot of other tools like unity. Visual Studio is used in this project in order to code the API which as written in C++.

3.1.5.2 OpenCV

OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products. Being a BSD-licensed product, OpenCV makes it easy for businesses to utilize and modify the code. Along with well-established companies like Google, Yahoo, Microsoft, Intel, IBM, Sony, Honda, Toyota that employ the library, there are many startups such as Applied Minds, VideoSurf, and Zeitera, that make extensive use of OpenCV. OpenCV's deployed uses span the range from stitching streetview images together, detecting intrusions in surveillance video in

Israel, monitoring mine equipment in China, helping robots navigate and pick up objects at Willow Garage, detection of swimming pool drowning accidents in Europe, running interactive art in Spain and New York, checking runways for debris in Turkey, inspecting labels on products in factories around the world on to rapid face detection in Japan [50].

OpenCV is used in this project to perform various operations on the frames read from the camera like thresholding, contour analysis, etc.

3.1.5.3 Android Studio

Android Studio is the official Integrated Development Environment (IDE) for Android app development, based on IntelliJ IDEA. On top of IntelliJ's powerful code editor and developer tools, Android Studio offers even more features that enhance productivity when building Android apps [51].

Android Studio is used in this project in order to create .so files which can be plugged into unity in order to interface Android with OpenCV and the developed API's C++ code.

3.1.5.4 Unity 3D

Unity3D is a powerful cross-platform 3D engine and a user-friendly development environment. Easy enough for the beginner and powerful enough for the expert; Unity should interest anybody who wants to easily create 3D games and applications for mobile, desktop, the web, and consoles. Unity has state of the art tools for building games for multiple platforms, all from the same editor. It also has inbuilt Virtual Reality development support with native support for Oculus. Additional SDKs (Software Development Kit) have to be installed in order to develop for other platforms like the Google Cardboard.

Unity3D is used in this project in order to create the demo Virtual Reality app for demonstrating the API.

3.2 System Analysis

The analysis phase is described as a problem-solving technique that decomposes a system into its component pieces for the purpose of the studying how well those component parts work and interact to accomplish their purpose [52]. It is a process of collecting and interpreting facts, identifying problems, and decomposition of a system into its components [53]. In this phase, the preliminary task was to understand the working and to research the availability of each the tools required to develop out project and the part that they play in the project.

Most of the analysis was done on the internet by reading various papers on computer vision and following tutorials for initializing and using the various tools. This phase also included studying existing systems which are similar to this project. Special attention had to be given on how each of the existing systems were created and on the advantages and drawbacks of each of them. Also, there are a number of algorithms for performing the same task in the computer vision world. Each of these algorithms had to be studied in order to choose the best one.

The project needed to be accomplished in a cost-effective method. So, cost analysis was performed by studying the various VR headsets and android phones with gyroscope that are available on the market. Similarly, the time and risks were analyzed before we proceeded with this project. Briefly stating, the analysis phase of this project can be broken down into the following phases:

- Feasibility Study
- Problem/Risk Analysis
- Requirement Analysis
- Logical Design

- Decision Analysis

3.3 System Design

Systems design is the process of defining the architecture, components, modules, interfaces, and data for a system such that they conform to the specified requirements. Systems design can be seen as the application of systems theory to product development [54] [55]. It implies a systematic and rigorous approach to design—an approach demanded by the scale and complexity of many systems problems.

The system designs used for building this project include Basic System Architecture, Algorithm, Flowchart, Context diagram, Data flow diagram and Use Case diagram.

3.3.1 Basic System Architecture

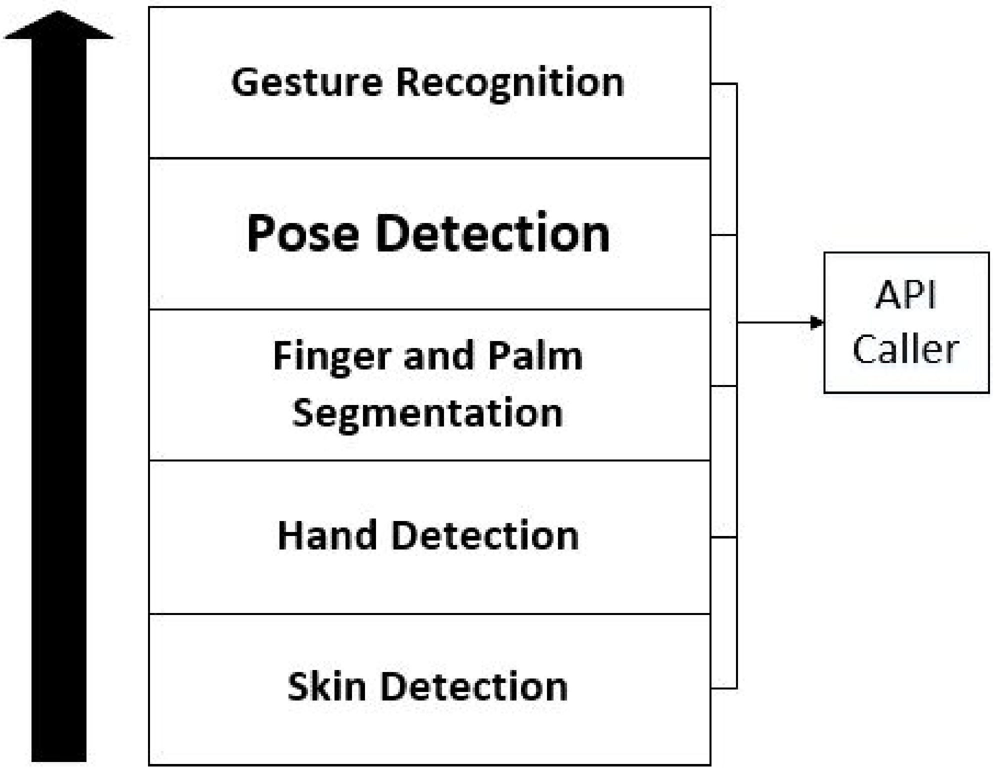


Figure 9 Layered Architecture of the System

The figure above shows the basic system architecture of the API. The system consists of five layers. The data flows from bottom to top where each higher layer has access to the output of all lower layers. The five layers are:

1. **Skin Detection Layer**: This layer is responsible for taking the frames obtained from the camera and detecting the pixels which may have skin on them. This way, the higher layers can only look at the pixels which are likely to be skin and save processing power.
2. **Hand Detection layer**: This layer uses the skin data in order to estimate which regions of skin might be hands.
3. **Finger and Palm Segmentation Layer**: This layer looks at the region of skin that could be a hand and tries to segment it into fingers and the palm.
4. **Pose Detection Layer**: This layer uses the finger and palm data in order to calculate the pose of the hand. The possible poses may be pre-defined using variables like the angle of the fingers, length of the fingers, etc., or a classifier might be trained and used.
5. **Gesture Recognition Layer:** This layer uses data calculated from a span of frames in order to determine the gesture that the detected hand is performing.

For example, the Pose Detection layer can access skin, hand as well as finger and palm segmentation data. The API Caller is the application that makes use of the API. This application can hook into any layer and access it's data. So, if the calling application requires only pose data but not gesture data, then we can skip the extra processing altogether.

3.3.2 Algorithm

3.3.2.1 Simple Skin Color Calibration

Step 1: Ask the user to place their hands on the rectangle

Step 2: Take a frame as input from the camera

Step 3: Draw a rectangle on the center of the frame

Step 4: Go to Step 2 until we have waited the desired number of frames

Step 5: Use the last recorded frame to calculate skin color value from the area inside the rectangle

3.3.2.2 Detection by Skin Color

Step 1: Use Calibration to get the user's skin color

Step 2: Take a frame as input from the camera

Step 3: Threshold the obtained frame using the calibrated color to get a binary image

Step 4: Reduce false positives in the binary image

Step 5: Analyze the binary image to find the region that might be the hand

Step 6: Analyze the hand region to find the fingers and palm

Step 7: Remove false positive for the fingertips

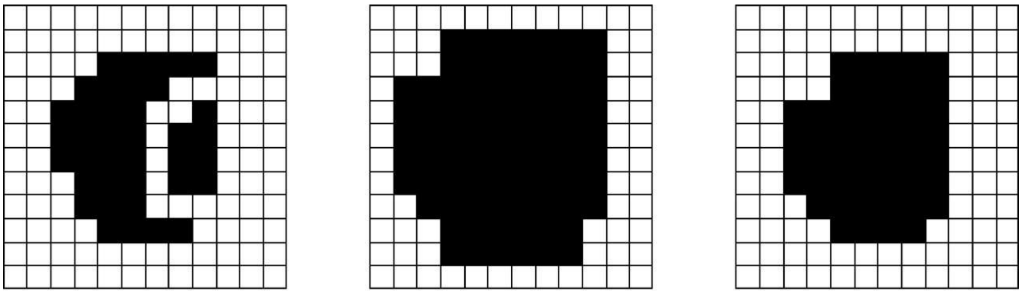
Step 8: Use the size and position of fingertips to guess the pose of the hand

Step 9: Put the obtained data in the queue for gesture analysis

Step 10: If there is enough data to detect gesture, process to get the gesture

Step 11: Go to Step 2 until the application is closed

There are multiple ways to implement each step of these algorithms. For instance, there are numerous ways to remove false positives from the binary image. One of the most common of such techniques is Image Opening and Image Closing. They use erode and dilate operations in order to reduce noise as well as fill up holes in the detected skin areas.



A. Binarizing may cause a faint floc to appear in two sections.

B. Dilate adds an outer layer of pixels to each floc, combining the two.

C. Erode removes the outer layer, making the size and shape more comparable to the original floc.

Figure 10 Dilation and Erosion Operations [56]

3.3.3 Flowchart

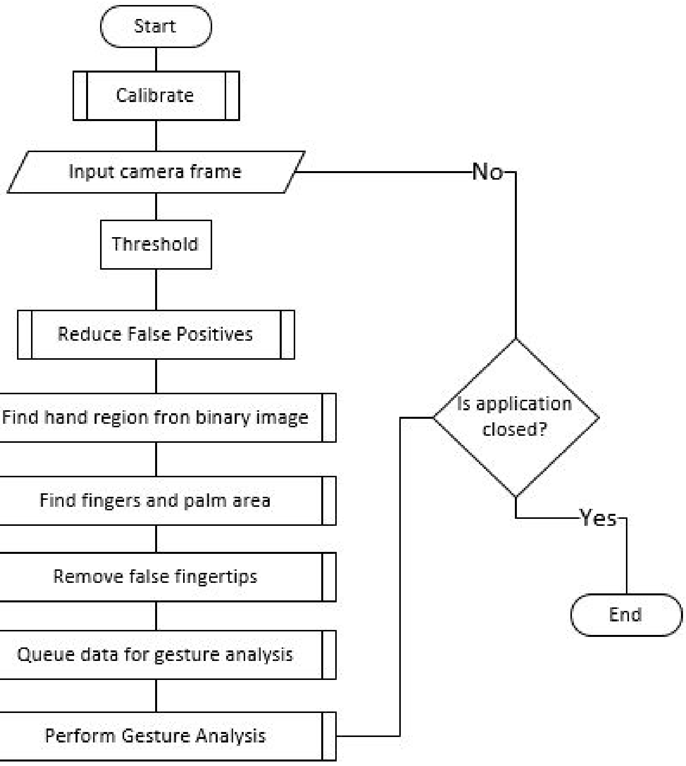
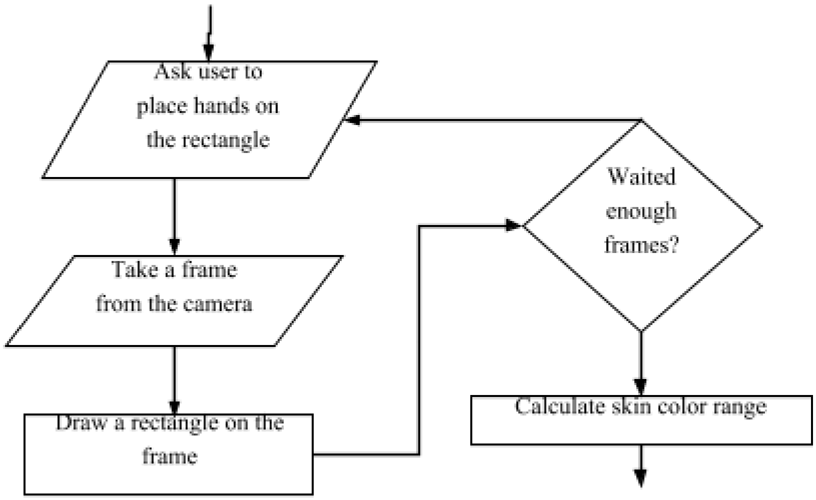


Figure 11 Flowchart of Detection by Skin Color



Start

End

Figure 12 Flowchart of Simple Skin Based Calibration

3.3.4 Context Diagram

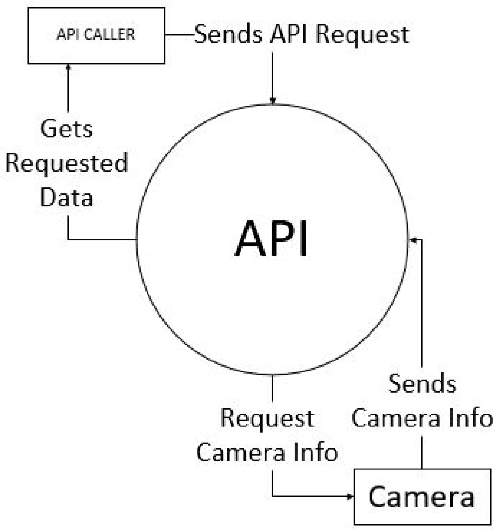


Figure 13 Context Diagram of the API

In figure 13, context diagram of the gesture recognition api is shown where API CALLER entity usually an application request a api function and API process it getting the required data from the camera. The processed data is forward to the API CALLER.

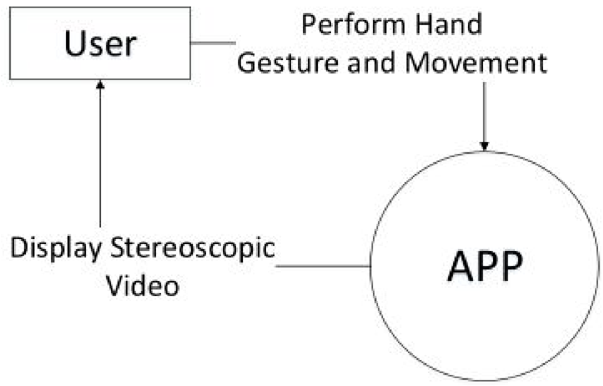


Figure 14 General Context Diagram of API Based App

In above figure 14, the user controls the app using hand movement and gesture recognition and the app displays stereoscopic video with processed gesture and movement.

3.3.5 Data Flow Diagram level 1

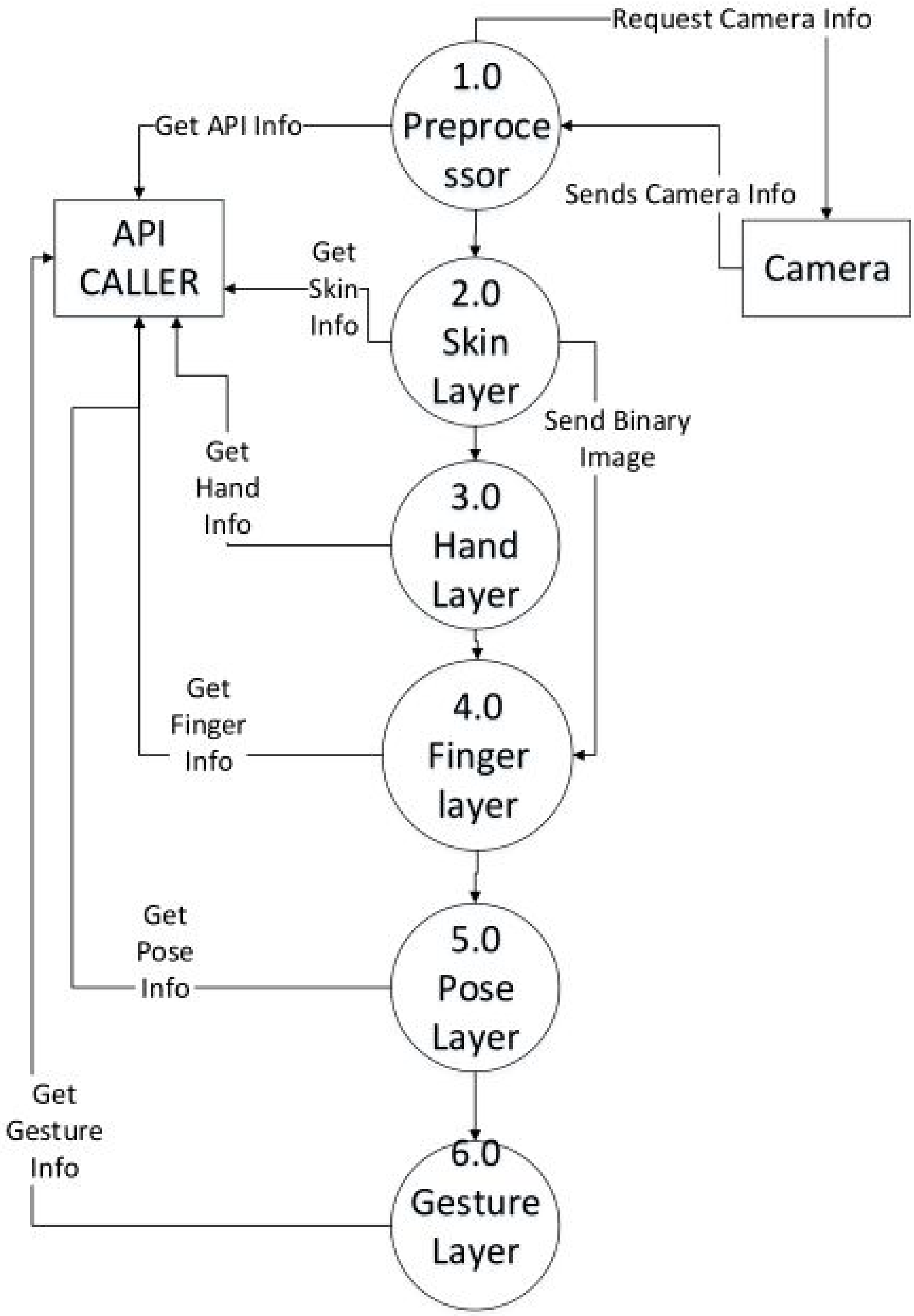


Figure 15 DFD 1 of the API

The figure 15 shows the data flow diagram of API which contains 6 sub-process and 2 major entity and they are listed below:

1.0 Pre-processor gets the camera feeds and forwards it to the API Caller. In between the forwarding process it also resizes and converts the color of the image.

2.0 Skin Layer gets the processed image and apply thresholding and perform skin operation and forwards the binary data.

3.0 Hand Layer gets the binary image and finds the contour. The processed data is forwarded to upper layer or API caller.

4.0 Finger Layer gets the binary image from hand layer and conducts major two operation i.e. fingertip locator and palm finder. These two operation sends different data finger info and palm info respectively.

5.0 Pose Layer compares the processed data with the defined pose data and gives pose info.

6.0 Gesture Layer also compares the info data with predefined gesture and gives gesture info.

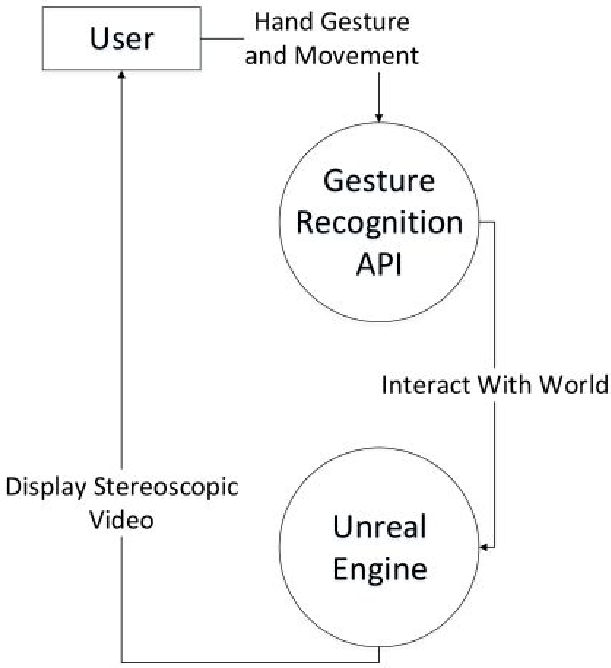


Figure 16 DFD1 of API Based App

Above figure 16, show the data flow diagram of the API based on app which consists of Gesture Recognition API and Unity engine. User sends their hand movement and gesture to the API which sends processed data to Unity Engine. The unity engine displays stereoscopic video as processed data.

3.3.6 Data Flow Diagram Level 2

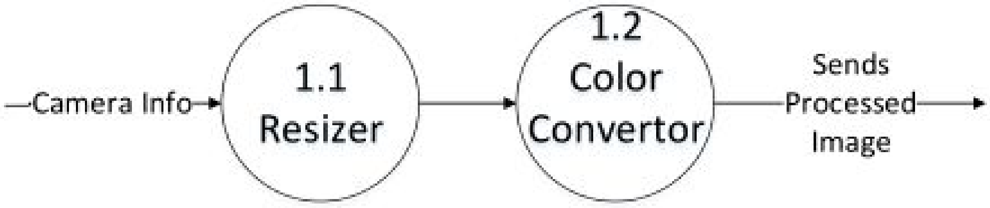


Figure 17 DFD 2 - 1.0 Preprocessor

Figure 17 shows two sub layers of preprocessor which described below:

1.1 Resizer has simple task of resizing the camera feed to fixed size which can be further processed by API.

1.2 Color Convertor converts the RGB image to HSV color model image.

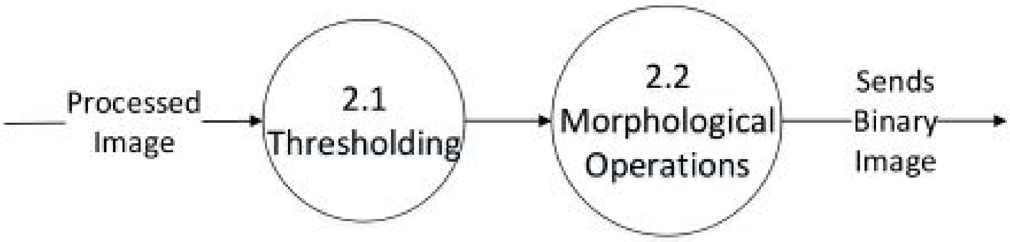


Figure 18 DFD 2 - 2.0 Skin Layer

The processed image from the preprocessor is sent to skin layer as shown in figure 18. The skin layer has two major function which are listed below:

2.1 Thresholding
2.2 Morphological Operations

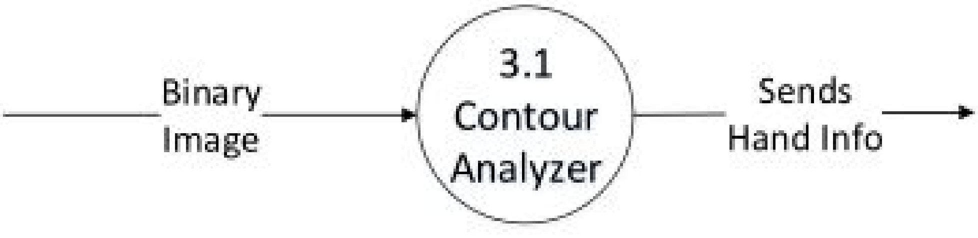


Figure 19 DFD 2 - 3.0 Hand Layer

In above figure 19, the hand layer gets the binary image and the contour analyzer has task of analyzing the contour of the image which separates the hand. The separated hand info is passed to next layer.

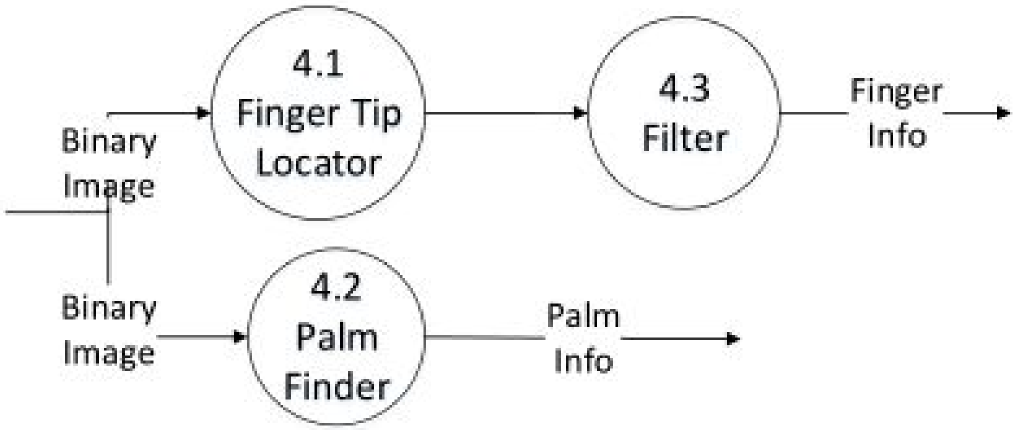


Figure 20 DFD 2 - 4.0 Finger Layer

The subsystem of the 4.0 finger layer is given in figure which consists of :

4.1 Finger Tip locator gets the binary image and locates the finger tip in it. It uses tip locator algorithm to locate the fingertip. Data is forwarded to another sub system filter.

4.2 Palm finder also gets the binary image and process it to show palm info to another layer.

4.3 Filter is used for stabilization of the fingertip. The data from finger tip locator is very unstable so a filter like Kalman filter must be applied

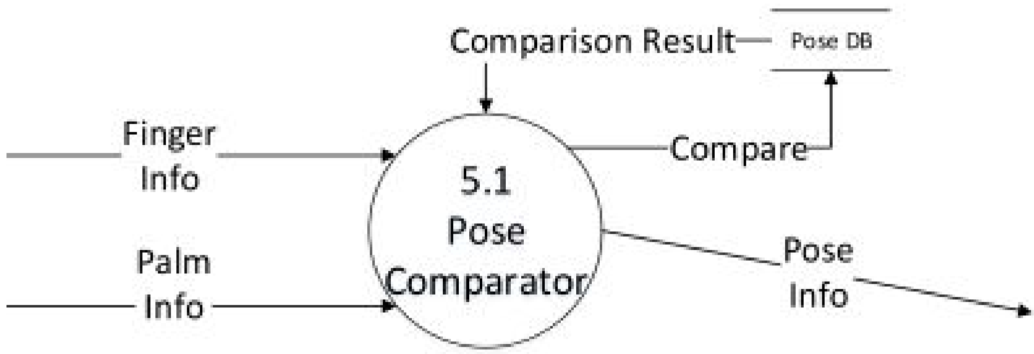


Figure 21 DFD 2 - 5.0 Pose Layer

The 5.0 pose layer has only one sub process i.e. 5.1 pose comparator as shown in figure 21 which gets finger and palm info from the previous layer. The info is then compared with predefined pose and the result is passed as pose info.

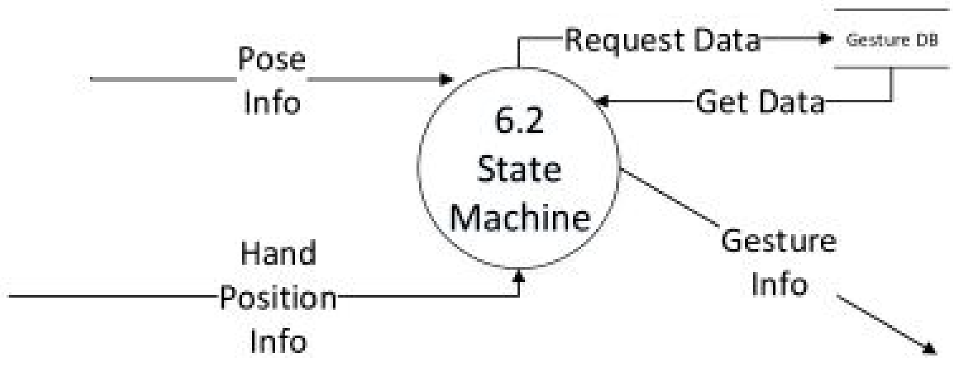


Figure 22 DFD 2 - 6.0 Gesture Layer

Figure 22 shows the subsystem of the 6.0 gesture layer where pose info and hand positon info is processed and compared with predefined gesture to produce gesture info.

3.3.7 Use Case Diagram

A use case diagram is a simple written description of how users perform tasks and interact with a system. It shows the system from a user's point of view and the system's behavior as it responds to a request [57]. It is basically a type of textual requirements specification that captures how a user will interact with a system to achieve a specific goal. The use case diagram usually contains an actor, basic flow, post conditions and processes [58].

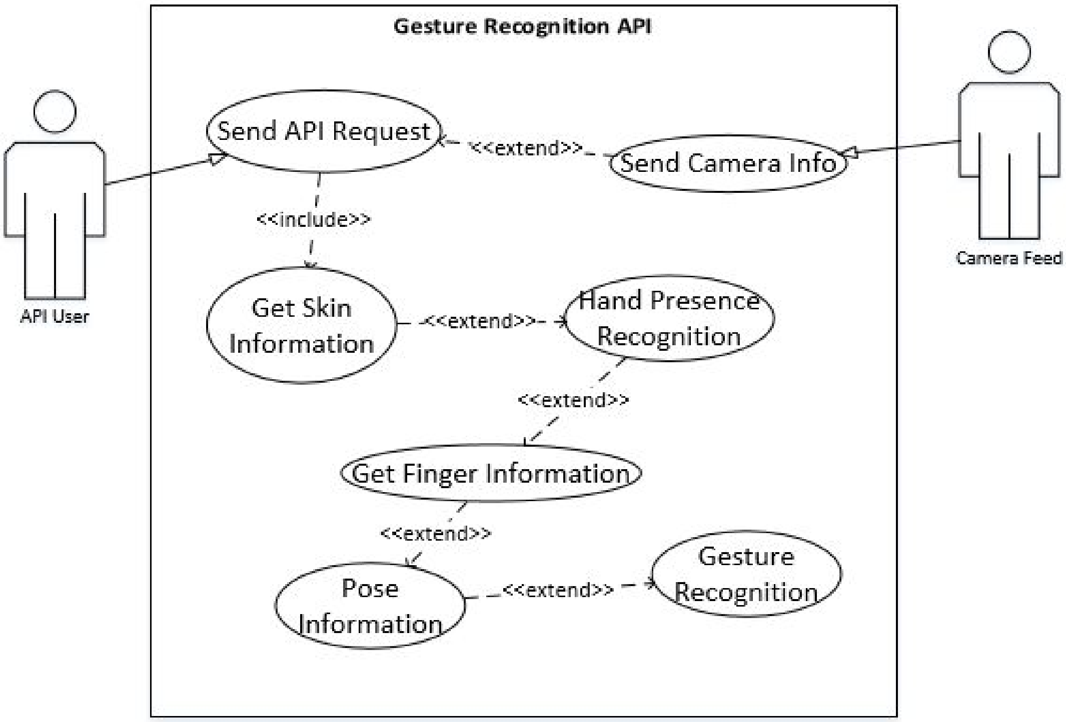


Figure: Use case diagram of the system

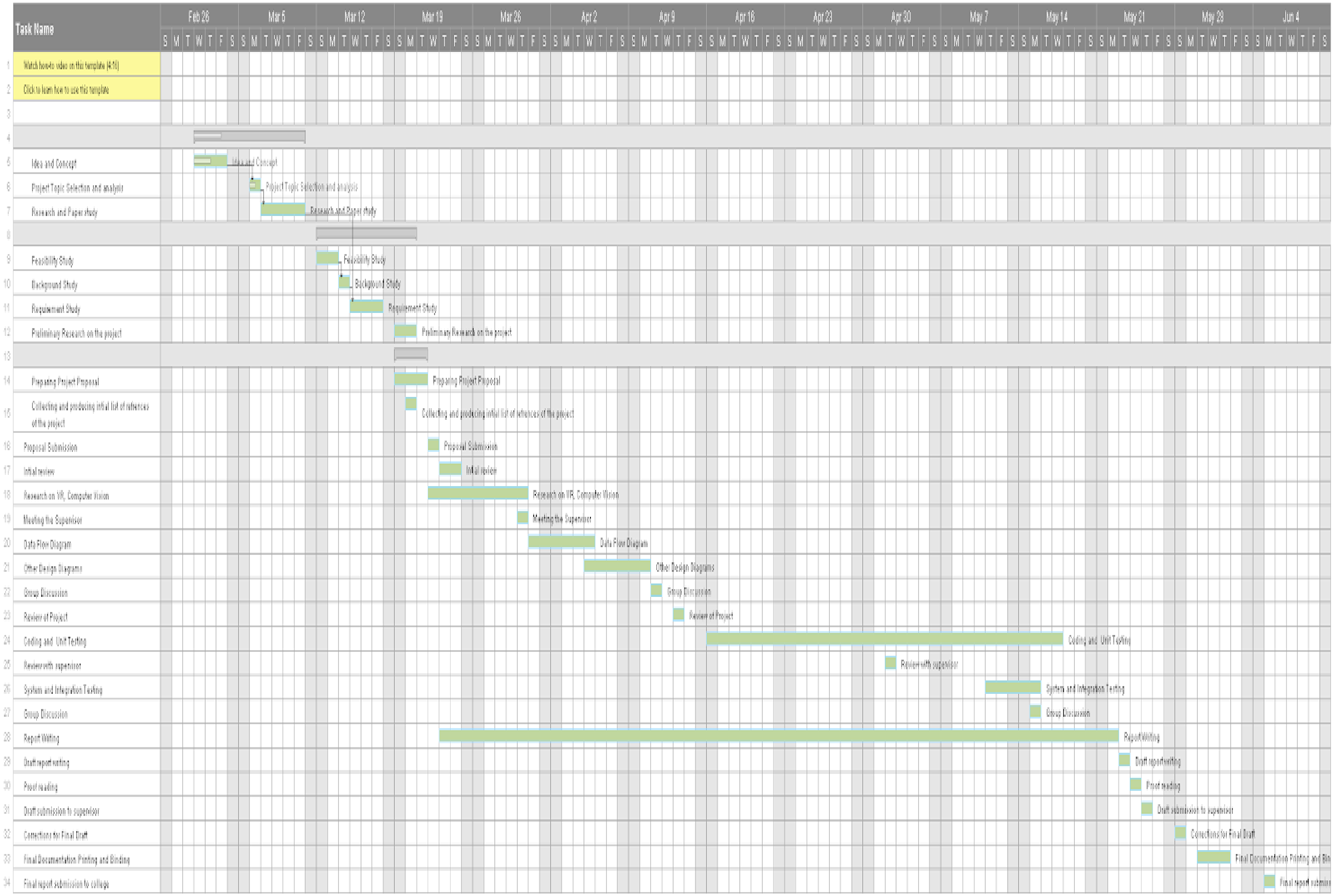
3.4 Project Schedule

Project schedule is the tool that helps to communicate what work needs to be performed, which resources of the organization will perform the work and the timeframes in which that work needs to be performed. The project was divided into different phases of work depending upon the requirement of the work. The phases were carried on in hierarchy as well as in parallel. The works were divided in groups and integrated later.

3.4.1 Time Schedule

Task Name	Start Date	End Date	Duration
Idea and Concept	03/01/17	03/03/17	3d
Project Topic Selection and analysis	03/06/17	03/06/17	1d
Research and Paper study	03/07/17	03/10/17	4d
Feasibility Study	03/12/17	03/13/17	2d
Background Study	03/14/17	03/14/17	1d
Requirement Study	03/15/17	03/17/17	3d
Preliminary Research on the project	03/19/17	03/20/17	2d
Preparing Project Proposal	03/19/17	03/21/17	3d
Collecting and producing initial list of references of the project	03/20/17	03/20/17	1d
Proposal Submission	03/22/17	03/22/17	1d
Initial review	03/23/17	03/24/17	2d
Research on VR, Computer Vision	03/22/17	03/30/17	7d
Meeting the Supervisor	03/30/17	03/30/17	1d
Data Flow Diagram	03/31/17	04/05/17	4d
Other Design Diagrams	04/05/17	04/10/17	4d
Group Discussion	04/11/17	04/11/17	1d
Review of Project	04/13/17	04/13/17	1d
Coding and Unit Testing	04/16/17	05/17/17	24d
Review with supervisor	05/02/17	05/02/17	1d
System and Integration Testing	05/11/17	05/15/17	3d
Group Discussion	05/15/17	05/15/17	1d
Report Writing	03/23/17	05/22/17	43d
Draft report writing	05/23/17	05/23/17	1d
Proof reading	05/24/17	05/24/17	1d
Draft submission to supervisor	05/25/17	05/25/17	1d
Corrections for Final Draft	05/28/17	05/28/17	1d
Final Documentation Printing and Binding	05/30/17	06/01/17	3d
Final report submission to college	06/05/17	06/05/17	1d

3.4.2 GANTT Chart



Task Name
Feb 26
Mar 5
Mar 12
Mar 19
Mar 26
Apr 2
Apr 9
Apr 16
Apr 23
Apr 30
May 7
May 14
May 21
May 28
Jun 4
Watch how-to video on this template (4:16)
Click to learn how to use this template
Idea and Concept
Project Topic Selection and analysis
Research and Paper study
Feasibility Study
Background Study
Requirement Study
Preliminary Research on the project
Preparing Project Proposal
Collecting and producing initial list of refences of the project
Proposal Submission
Initial review
Research on VR, Computer Vision
Meeting the Supervisor
Data Flow Diagram
Other Design Diagrams
Group Discussion
Review of Project
Coding and Unit Testing
Review with supervisor
System and Integration Testing
Group Discussion
Report Writing
Draft report writing
Proof reading
Draft submission to supervisor
Corrections for Final Draft
Final Documentation Printing and Binding
Final report submission to college

3.5 System Testing

Implementation of a project remains fruitful if no error befalls during operation. Ensuring the lack of error is only possible if the system is properly tested and does not fail in different test approaches. Testing a computer vision system varies from testing normal, everyday software. It is difficult to automate computer vision tests. It is often a tricky task to determine what is causing errors in the system. Because of the complexity of testing computer vision systems, unit tests prove to be the best bet. The system was tested by testing individual units first and then testing how they integrate with each other. Each layer of the architecture was tested separately to ensure that they give valid output which can be reused by the other layers. Various testing approaches were implemented throughout the project and specific tests were carried out at the completion phase. The tests carried out in this project are described below.

3.5.1 Component test

Component tests, also known as unit tests are used to test small, individual parts of a project in isolation. Here, each individual component goes through a test phase that verifies that the component operates normally. The component test was carried out for the components inside each of the five layers of the system. Each of these layers were processed independently of the others in order to test the system.

Test cases:

1) **Color Space Test**

When extracting the skin regions from a frame, we use the user's skin color which is obtained through calibration. Before even carrying out calibration, it is important to choose the correct color model from the range of models available like RGB, YCbCr, HSV, etc. It was found that the YCbCr and HSV color models perform the best job in detecting skin color.

2) Calibration Test

The calibration routine was carried out multiple times by varying the number of samples taken from the user's skin as well as the range on which the thresholding was done. The best results were found by using a wide range for the Saturation and Value components of the HSV color model compared to the Hue component.

3) False Positive Removal Test

As discussed above, erosion and dilation can be used to remove the false positive skin from the threshold/binary image. On top of that, techniques such as Median Blur, Gaussian Blur, etc. can be used in order to remove false positives. The different combination of all of these techniques were tested thoroughly in order to determine the best one.

4) Hand Recognition Test

This test was carried out in order to verify that the system correctly identifies the hand skin region from multiple skin regions obtained on the threshold/binary image.

5) Fingertip Recognition Test

This test was carried out in order to verify that the system correctly finds the fingertips in the recognized hand area. This required the use of multiple optimization techniques.

6) Finger Count Test

This test was carried out in order to verify that the system correctly identifies and counts the number of fingers in a sequence of input frames.

7) Pose Test

This test was carried out in order to verify that the system can correctly identify a set of predefined poses.

8) Gesture Test

This test was carried out in order to verify that the change in the fingertip positions and number of fingers was correctly identified by the system and mapped into a gesture.

3.5.2 Integrated Test

The integrated test is a specialized testing carried out by merging various component tests as one. In this phase, each of the individual components and layers were connected together and tested. This test was required to analyze the coordinated operation of the different layers and processes in the system.

1) Calibration and Thresholding Test

This test was carried out to verify that the calibrated skin color values could effectively serve the Skin Layer to properly threshold the image and recognize the parts that contain skin.

2) Skin and Finger Test

This test was carried out to verify that the Skin region obtained from the skin layer could effectively be used to recognize the presence and position of the fingertips in the system.

3) Finger and Pose Test

This test was carried out to verify that the finger position and count obtained from the

Finger Layer could effectively be used to recognize a predefined set of hand poses

4) Pose, Position and Gesture Test

This test was carried out to ensure that the Gesture Layer can take proper input from the hand and pose layers in order to identify a pre-defined set of gestures.

3.5.3 Functional Test

A functional test, as specified by its name, is used to test the working of the system components as whole. Similar to integrated test, a functional test varies with the fact that it is able to inspect whether a system of integrated components is able to function well or not. In the functional test, the system was checked by streaming a camera feed and attempting to recognize the fingertips from the feed.

3.5.4 Operational test

The operational test is a final test for acceptance. Also referred to as system test, this type of test is carried out at the end of the project. In this test, the actual operation is monitored by integrating all of the components of the system and verifying their operation. Once the system responds positively, the system is finalized and ready for the implementation phase.

CHAPTER 4: RESULT ANALYSIS

4.1 Screenshots

The screenshots of the main parts of the system are shown below:

4.1.1 Color Based Calibration

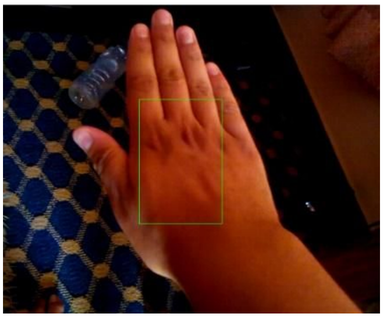


Figure 23 Calibration

The above figure shows how the user is asked to place their hands inside the designated rectangle and the skin color of the user is extracted for use in further processing.

4.1.2 Color Based Thresholding

Figure 24 Thresholding | Left - Original Image, Right - Thresholded Image

The above figure shows how an input image was thresholded by using data taken from the calibration phase. It still has some false positives. This result can be improved by taking more number of samples from the calibration phase and by using adaptive or dynamic skin recognition.

4.1.3 Hand Recognition by Convex Hull

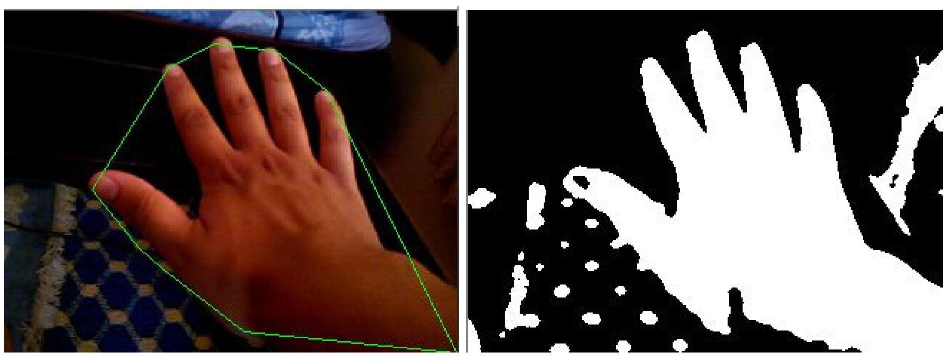


Figure 25 Hand Recognition by Convex Hull

The above figure shows how the user's hands are recognized by finding and drawing a convex hull around it.

4.1.4 Hand Recognition by Convexity Defect Analysis

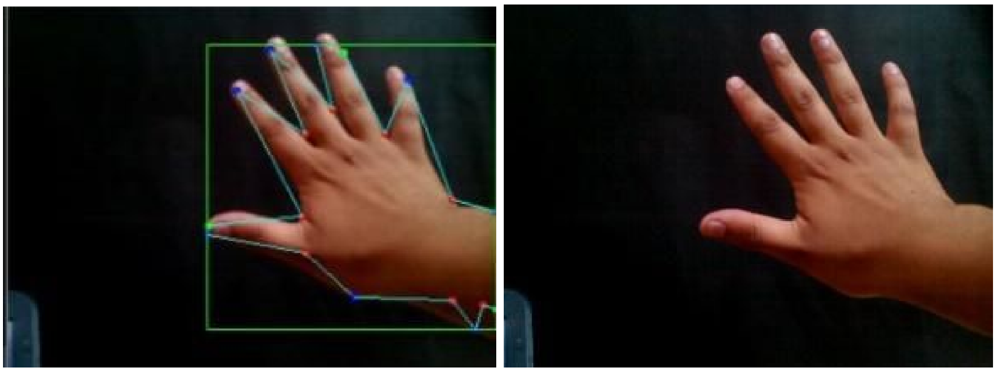
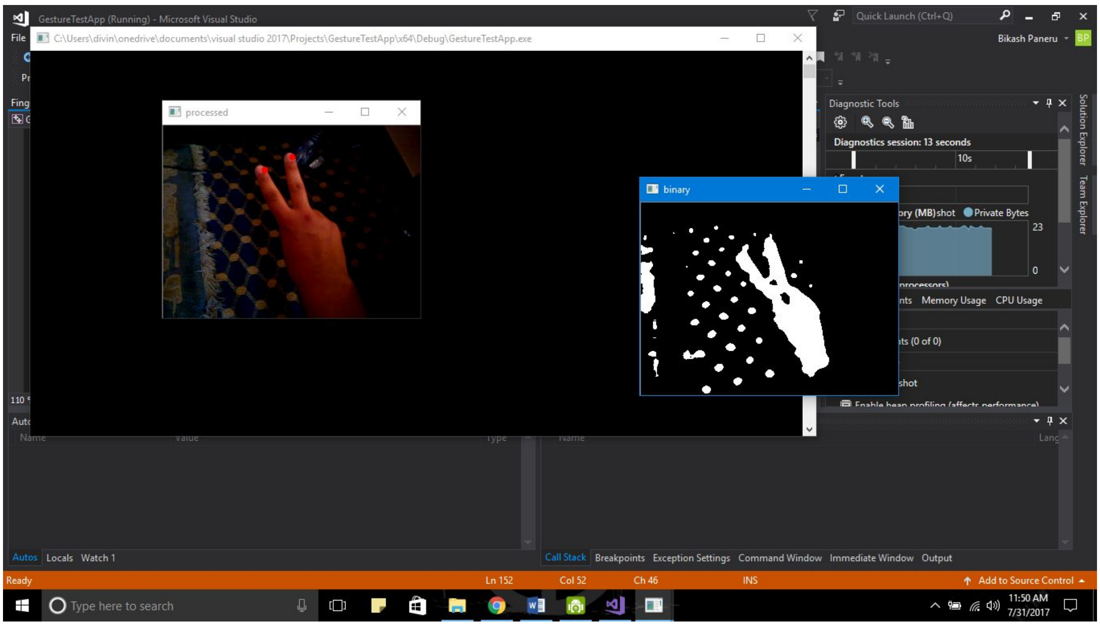


Figure 26 Hand Recognition by Convexity Defect Analysis

The above figure shows how the user's hands are recognized by using the convexity defect analysis technique.

4.1.5 Fingertip Recognition



The above figure shows how the user's hands are analyzed and the positions of the fingertips are generated.

4.1.6 Pointer Position Estimation

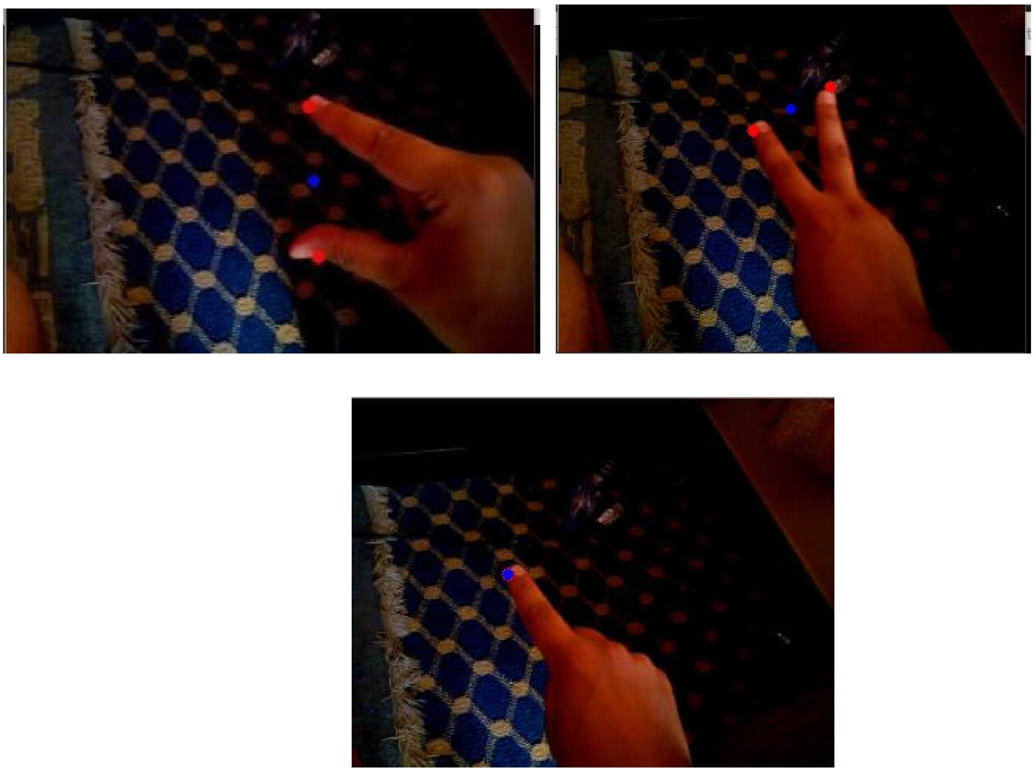


Figure 27 Pointer Estimation

The above figures show how the position of the fingers are analyzed in order to estimate the position of the pointer.

4.2 Critical Analysis

Gesture recognition, along with facial recognition, voice recognition, eye tracking and lip movement recognition are components of what developers refer to as a perceptual user interface (PUI). The goal of PUI is to enhance the efficiency and ease of use for the underlying logical design of a stored program, a design discipline known as usability.

In personal computing, gestures are most often used for input commands. Recognizing gestures as input allows computers to be more accessible for the physically-impaired and makes interaction more natural in a gaming or 3-D virtual reality environment. Hand and body gestures can be amplified by a controller that contains accelerometers and gyroscopes to sense tilting, rotation and acceleration of movement -- or the computing device can be outfitted with a camera so that software in the device can recognize and interpret specific gestures. A wave of the hand, for instance, might terminate the program.

With the increase in computer usage in the world, computer aided learning is being more and more popular. Many studies of e-learning programs have concluded that the key to ensuring successful outcomes is to blend more traditional classroom approaches with those that use technology. eLearning approaches should be treated as a powerful tool that teachers can use, but teachers need to learn how student learning changes with e-learning, and how to alter their teaching methodologies with pedagogical approaches that take advantage of the opportunities afforded by e-learning. A blended approach mixing face-to-face classroom methods with technology mediated activities seems to provide the highest learning outcomes. In African countries where secondary teachers have little prior experience with computers or similar technologies, all this can be a significant undertaking. In-service and pre-service e-learning teacher training programs have been designed and tested, however, with these goals in mind.

This project is created to bridge the gap between the traditional education system and enrolment of ICT in education in current situation. The usage of computer in Nepal is low compare to Global statistics. The major challenge of this project is to provide the reach of this application in every area of the country. We initially reached on the technology used in the current education process. After through research in e learning methodology, it was decided that virtual reality is emerging technology and should be practiced in developing nation like ours. We wanted to build a system that was easy and clear to use and also is inexpensive to implement. Technology used must be low cost operating. After vague analysis, we found that Unity VR development engine for the virtual reality interface. The engine was free to buy and easy to build and deploy application in it.

The project required proper hardware with low operating cost. Thus, Google cardboard was a great fit in our requirement. Input in the box was given by hand to reduce the cost of data gloves and other technology.

Figure 28 Comparison between different VR Headsets [59]

As per hardware selected, the project was developed in Android platform. Android OS is popular mobile operating system because of its portability, and huge market share. Google cardboard was used as it is inexpensive and easy to build. For input purpose, OpenCV – computer vision library is used as the library is free and open to use. The library uses a simple camera and gives ability to use the power of computer vision. Computer vision is an interdisciplinary field that deals with how computers can be made for gaining high-level understanding from digital images or videos. It is used to manipulate images and camera feed of smartphones and manipulate them to process them according to need of the user. It is basically used to hand recognition in the system. Many algorithms and principles of hand and gesture recognition was analyzed.

Unity and Google SDK for VR provided the virtual reality environment to build application for android. The system architecture is that an application uses Gesture Recognition API. The app request gesture input from API and application process the gesture request.

The expensive VR education module cannot be used in our country. The infrastructure of the nation is not as good compare to other developed nation. This system makes use of emerging technology is low cost. The major goal of this system is make technology available to all regarding their economic condition.

Implementation of this project is entirely based in the Head Mounted Device, here it is a Virtual Reality Box. User needs to calibrate their hand with the system. The system that implements the Gesture Recognition API would now take the user's hand as input for the system. The API will now process the request call from the application. The image is provided to the API via the system's camera. The algorithm now starts to work on the image. The recognize hand gesture is forwarded to application where it is manipulated to process the request of the user.

The application provided in the demo has used the API and implemented a simple eLearning application in which student can play with virtual 3D model of animals and know many things about it. The application would make student clearer about animals by being closer to it. They can manipulate them and know how they eat, run or walk. With virtual reality, students can see things which they can't dream of. Example, with this application student can analyze a marine animal without being in a real world.

This Gesture Recognition API and its implementation in education makes the technology reach to remote corners of the nation. It may not totally replace the curriculum but would make a tremendous advancement in the field of education. The creative and imagination power of the student can be changed as they learning by observing. The API would be beneficial for many practical where the instruments are highly expensive. The API will make the use of user hand making the VR more and more of a reality. This API and its application would help many people by making the technology available at less cost and also making the use of available technology to fullest.

4.3 Limitations and future enhancements

4.3.1 Limitations

No matter how a system is analyzed and predicted to be foolproof, there always remains a room for improvement. No project can be developed with 100% accuracy or zero limitations. There are some limitations in this project that emerged and could not be covered in this phase. Some of the limitations are:

1. Proper Lights must be maintained for Hand Recognition.
2. VR application is limited to only Android platform.
3. Some hand gesture may not be recognized.

4.3.2 Future Enhancements

Regardless of the limitations, some minor enhancements can be carried out to further enhance this project. In our view, some of the possible enhancements are stated in the following section:

1. Increasing the VR field to each and every platform.
2. Using gaze gesture for another source of input.
3. Voice Control support in the API

4.4 Conclusion

In the field of computer science, every other day there is a rise of new technology. We must realize this technology and start working on them. A huge research in these kinds of technology must be done. We must not be traditional but must think out of the box. Gesture recognition and Virtual Reality may be one of the emerging technology in the current world. A great research and with its application can help the nation in each and every sector.

We simply have shown how gesture based virtual reality application can be in the field of education. The tradition we follow i.e. paper based education cannot be replaced but we can alter some of their process so that we can use them for our benefits. A student will also give their full attention to the classes because the animation, the 3d models is attracting them. This project aims to change the process of learning by making it more fun, easy and interactive.

Education being the backbone of the nation is still the same that was 30 years ago. Yes, the infrastructure like physical buildings, facilities, etc. has been changed but the principle we follow is the same. So, in this context this project helps in bringing the change in the education system.

The concept of this whole project was to involvement of ICT in the present scenario of education system. The gesture based application is going to help the people to think more and do more. The application of this project is endless. It can be used to view and analysis the sub atomic practices of an atom. It can used to develop the a hand based chemical laboratory. Not only it makes the process of learning or knowing fun but also increases the safety of the user. One can simple know about the about sea corals without going for deep sea diving. Even doctors and engineers would use the project to test or use it as a simulator.

Lastly, the project was built to decrease the technology gap between our nation and rest of the world and also to change the process of learning. The primary objective of the project was completed successfully. But, further enchantment can be done to increase the use of the project. Reducing the limitation and further increasing the feature will definitely help in increasing the effectiveness of the project. The outcome of all these hard works and patience has now leaded to successfully demonstration of new technology and its application in education.

REFERENCES

[1] K. I. Manavadariya, A. J. Faldu, P. C. Shukla and D. J. Patel, "Computer Based Human Gesture Recognition With Study Of," *IOSR Journal of Computer Engineering,* vol. 11, no. 2, pp. 41-50, 2013.

[2] S. J. J. Kim, "A User Study Trends in Augmented Reality and Virtual Reality Research: A Qualitative Study with the Past Three Years of the ISMAR and IEEE VR Conference Papers," in *IEEE*, Adaejeon, South Korea, 2012.

[3] "Microsoft Hololens," [Online]. Available: https://compass-ssl.surface.com/assets/f5/2a/f52a1f76-0640-4a37-a650-51b0902f8427.jpg?n=Buy_Panel_1920.jpg. [Accessed 25 7 2017].

[4] Microsoft, "The leader in Mixed Reality technology | HoloLens," Microsoft, [Online]. Available: https://www.microsoft.com/en-us/hololens. [Accessed 25 7 2017].

[5] "Facebook Buys Pebbles Interfaces to Join Oculus," [Online]. Available: https://static2.i4u.com/sites/default/files/images/content_images/pebbles-interfaces-1.jpg. [Accessed 30 7 2017].

[6] "Pebbles Interfaces," [Online]. Available: http://www.startbase.co/uploads/Startups/097abc3e91202c19ab8a6788c7c8fc11.jpg. [Accessed 25 7 2017].

[7] Crunchbase.com, "https://www.crunchbase.com/organization/pebbles-interfaces#/entity," [Online]. Available: https://www.crunchbase.com/organization/pebbles-interfaces#/entity. [Accessed 25 7 2017].

[8] "Microsoft Kinect Sensor," [Online]. Available: https://www.generationrobots.com/img/cms/Kinect-sensor.png. [Accessed 25 7 2017].

[9] Microsoft, "Kinect for Xbox One | Xbox," Microsoft, [Online]. Available: http://www.xbox.com/en-US/xbox-one/accessories/kinect. [Accessed 25 7 2017].

[10] "Android One Mobile Price List in India," [Online]. Available: http://www.mysmartprice.com/mobile/pricelist/android-one-mobile-price-list-in-india.html. [Accessed 25 7 2017].

[11] P. Hands and C. J. Read, "True stereoscopic 3D cannot be simulated by shifting 2D content off the screen plane," *Science Direct,* vol. 48, p. 35–40, 2017.

[12] M. Deering, "Data complexity for virtual reality: where do all the triangles go?," *Virtual Reality Annual International Symposium,* 1993.

[13] B. Alicea, "Relativistic virtual worlds: an emerging framework," Michigan State University, 2011.

[14] M. Alcaniz, J. Lozano and B. Rey, "Technological background about virtual reality," in *Cybertherapy*, Amsterdam, IOS Press, 2006.

[15] W. Sherman and A. Craig, Understanding Virtual Reality: interface, application, and design, San Francisco, 2003.

[16] S. K. M. L. Lucchese, "Color Image Segmentation: A State-of-the-Art Survey," *Indian national science academy,* vol. 67, no. 2, pp. 207-221, 2001.

[17] A. K. Władysław Skarbek, "Color Image Segmentation: A Survey," Berlin, Germany, 1994.

[18] H.-S. Yeo, B.-G. Lee and H. Lim, "Hand Tracking and Gesture Recognition System for Human-Computer Interaction using Low-cost Hardware," *Multimedia Tools and Applications (MTAP),* vol. 74, no. 8, 2015.

[19] K. Sgouropoulos, E. Stergiopoulou and N. Papamarkos, "A Dynamic Gesture and Posture Recognition System," *Journal of Intelligent & Robotic Systems,* vol. 76, no. 2, p. 283–296, 2014.

[20] N. A. Ibraheem, R. Z. Khan and M. M. Has, "Comparative Study of Skin Color based Segmentation Techniques," *IJAIS,* vol. 5, no. 10, 2012.

[21] A. B. D. C. Son Lam Phung, "Skin Segmentation Using Color Pixel Classification: Analysis And Comparison," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 27, no. 1, pp. 148-154, 2005.

[22] M. N. S. H. F. S. A. Jean-Christophe Terrillon, "Comparative Performance of Different Skin Chrominance Models and Chrominance Spaces for the Automatic

Detection of Human Faces in Color Images," in *IEEE the Fourth International Conference on Automatic Face and Gesture Recognition*, 2000.

[23] A. Dhawan and V. Honrao, ""Implementation of Hand Detection based Techniques for Human Computer Interaction," *Int. J. Comput. Appl.,* vol. 72, no. 17, pp. 6-13, 2013.

[24] "Robust vision-based hand tracking using single camera for ubiquitous 3D gesture interaction," *3D User Interfaces (3DUI), 2010 IEEE Symposium,* 2010.

[25] S. J. Nowlan and J. C. Platt, "A Convolutional Neural Network Hand Tracker," in *NIPS'94 Proceedings of the 7th International Conference on Neural Information Processing Systems*, 1995.

[26] Q. Chen, N. D. Georganas and E. M. Petriu, "Hand Gesture Recognition Using Haar-Like Features and a Stochastic Context-Free Grammar," *IEEE Transactions on Instrumentation and Measurement,* vol. 57, no. 8, 2008.

[27] A. Tang, K. Lu, Y. Wang, J. Huang and L. Houqiang, "A Real-Time Hand Posture Recognition System Using Deep Neural Networks," *ACM Transactions on Intelligent Systems and Technology (TIST),* vol. 6, no. 2, 2015.

[28] D. A. Verma R., "Vision based Hand Gesture Recognition Using finite State Machines and Fuzzy Logic," in *International Conference on Ultra-Modern Telecommunications & Workshops*, 2009.

[29] P. T. J. J. Nguyen D.D., "Fingertip Detection with Morphology and Geometric Calculation," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, St. Louis, USA, 2009.

[30] Z. H. Xu Z., "Vision-based detection of dynamic gesture," *International Conference on Test and,* pp. 223-226, 2009.

[31] J. K. N. K. D. Choi, "Morphological Gesture Recognition Algorithm," in *Proceeding of IEEE region 10th international conference on Electrical and Electroic Technology*, Coimbra, Portugal, 2001.

[32] B. G. Trivino G., "Linguistic description of human body posture using fuzzy logic and several levels of abstraction," in *IEEE conference on Computational Intelligence for measurement systems and applications*, Ostuni, Italy, 2007.

[33] K. Kholi, J. P. Singh and A. Kumar, "Motion Detection Algorithm," *The International Journal of Computer Science & Applications (TIJCSA)* , vol. 1, no. 12, 2013.

[34] R. H. Nolker C., "Visual Recognition of Continuous Hand Postures," *IEEE Transactions on neural,* vol. 13, no. 4, pp. 983-994, 2002.

[35] P. N. Stergiopoulou E., "Hand gesture recognition using a neural network shape fitting technique," *Engineering Applications of Artificial Intelligence,* vol. 22, no. 8, pp. 1141-1158, 2009.

[36] S. Jang, J. M. Vitale, R. W. Jyung and J. B. Black, "Direct manipulation is better than passive viewing for learning anatomy in a three-dimensional virtual reality environment," *Computers & Education,* vol. 106, pp. 150-165, 2017.

[37] N. R. Valdez, M. V. Rivera and J. P. Pabico, "Experiences in Implementing an ICT-Augmented Reality as an Immersive Learning System for a Philippine HEI," *Asia Pacific Journal of Education, Arts and Sciences,* vol. 2, no. 2, 2015.

[38] A. Sari and H. Mahmutoglu, "Potential Issues and Impacts of ICT Applications through Learning Process in Higher Education," *Procedia - Social and Behavioral Sciences,* vol. 89, pp. 585-592, 2013.

[39] "THE DEVELOPMENT OF EDUCATIONAL TECHNOLOGY POLICIES (1996-2012) LESSONS FROM CHINA AND THE USA," *International Education Studies,* vol. 8, no. 6, 2015.

[40] M. S. H. Khan, M. Hasan and C. K. Clement, "Barriers to the Introduction of ICT into Education in Developing Countries: The Example of Bangladesh," *International Journal of Instruction,* vol. 5, no. 2, pp. 61-80, 2012.

[41] R. Balaji, F. Al-Mahri and R. Malathi, "A Perspective Study on Content Management in ELearning," eprint arXiv:1605.02093, 2016.

[42] M. &. P. M. Tayebinik, "Blended Learning or E-learning?," *International Magazine on Advances in Computer Science and Telecommunications (IMACST),* vol. 3, no. 1, pp. 103-110, 2012.

[43] "Association of Project Management," [Online]. Available: https://www.apm.org.uk/WhatIsPM. [Accessed 26 06 2017].

[44] Project Management Institute, A Guide to the Project Management, 2013.

[45] S. J. Zeil, "Software Development Process Models," 19 10 2015. [Online]. Available: https://www.cs.odu.edu/~zeil/cs350/f15/Public/processModels/. [Accessed 26 7 2017].

[46] "Iterative Model," [Online]. Available: https://www.tutorialspoint.com/adaptive_software_development/images/iterations.jpg. [Accessed 26 7 2017].

[47] ISBM Exam Certification, "What is Incremental model- advantages, disadvantages and when to use it?," [Online]. Available: http://istqbexamcertification.com/what-is-incremental-model-advantages-disadvantages-and-when-to-use-it/. [Accessed 26 7 2017].

[48] Tutorialspoint, "SDLC Iterative Incremental Model," [Online]. Available: https://www.tutorialspoint.com/adaptive_software_development/sdlc_iterative_incremental_model.htm. [Accessed 26 7 2017].

[49] Microsoft, "Visual Studio IDE," [Online]. Available: https://www.visualstudio.com/. [Accessed 26 7 2017].

[50] OpenCV, "About - OpenCV Library," [Online]. Available: http://opencv.org/about.html. [Accessed 26 7 2017].

[51] Google Developers, "Meet Android Studio," [Online]. Available: https://developer.android.com/studio/intro/index.html. [Accessed 26 7 2017].

[52] Lonnie D. Bentley, "Systems Analysis and Design for the Global Enterprise," p. 160.

[53] Tutorialspoint, "System Analysis and Design Overview," [Online]. Available: https://www.tutorialspoint.com/system_analysis_and_design/system_analysis_and_design_overview.htm. [Accessed 26 7 2017].

[54] National Communications System, "Federal Standard 1037C," National Communications System, 1996.

[55] United States Department of Defense, "Dictionary of Military and Associated Terms".

[56] *Journal of Sedimentary Research.*

[57] "Use Cases," [Online]. Available: https://www.usability.gov/how-to-and-tools/methods/use-cases.html.

[58] Laura Brandenburg, "How to Write a Use Case," [Online]. Available: http://www.bridging-the-gap.com/what-is-a-use-case/. [Accessed 26 06 2017].

[59] Slideshare, [Online]. Available: https://image.slidesharecdn.com/3xuyowo4s0une2proypd-signature-7873c75031ac31677418d06ca1135ba58cf1db5a062456d83a04778887726afe-poli-151012220105-lva1-app6891/95/a-literal-360degree-view-of-your-data-with-google-cardboard-5-638.jpg?cb=1444690198. [Accessed 27 7 2017].