NATURAL USER INTERFACE

MAIN PROJECT REPORT

Submitted by

ANU PAULY

RENITTO JOSE E

SHON JOICE T

SREELAKSHMY N

TOM JOSE C

in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY (B.TECH)

in

COMPUTER SCIENCE & ENGINEERING

of

UNIVERSITY OF CALICUT

Under the guidance of

Ms Kumary R Soumya

MAY 2012

Department of Computer Science & Engineering

JYOTHI ENGINEERING COLLEGE, CHERUTHURUTHY

THRISSUR 679 531

NATURAL USER INTERFACE

MAIN PROJECT REPORT

Submitted by

ANU PAULY

RENITTO JOSE E

SHON JOICE T

SREELAKSHMY N

TOM JOSE C

in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY (B.TECH)

in

COMPUTER SCIENCE & ENGINEERING

of

UNIVERSITY OF CALICUT

Under the guidance of

Ms Kumary R Soumya

MAY 2012

Department of Computer Science & Engineering

JYOTHI ENGINEERING COLLEGE, CHERUTHURUTHY

THRISSUR 679 531

Department of Computer Science & Engineering

JYOTHI ENGINEERING COLLEGE, CHERUTHURUTHY

THRISSUR 679 531

MAY 2012

BONAFIDE CERTIFICATE

Certified that this project report **" NATURAL USER INTERFACE "** being submitted in partial fulfillment of the requirements for the award of degree of **Bachelor of Technology** of **University of Calicut** is the bonafide work of **ANU PAULY, RENITTO JOSE E, SHON JOICE T, SREELAKSHMY N, TOM JOSE C** , who carried out the project work under our supervision.

Prof.ASWATHY WILSON	Ms.KUMARY R SOUMYA
HOD	**PROJECT GUIDE**
Dept. of CSE	Assistant Professor
	Dept. of CSE

ACKNOWLEDGEMENT

We take this opportunity to express our heartfelt gratitude to all respected personalities who had guided, inspired and helped us in the successful completion of this project.

First and foremost, we express our thanks to **The Lord Almighty** for guiding us in this endeavour and making it a success.

We are thankful to our Principal **Dr Gylson Thomas** and the Management for providing us with excellent lab and infrastructure facilities.

Our sincere thanks to the Head of the Department of Computer Science and Engineering, **Prof. Aswathy Wilson** for her valuable guidance and suggestions.

We would like to express our deepest gratitude to **Ms.Kumary R Soumya** for her valuable contributions and guidance.

Last but not least, we thank all our teaching and non teaching staffs of Department of Computer Science & Engineering, and also our friends for their immense support and help in all the stages for the development of the project.

ABSTRACT

The Augmented reality has become a promising and fast growing application of computer graphics over the course of the years. It refers to a system in which the physical surroundings of a person are mixed with real time computer generated information creating an enhanced perception of surrounding environment. In this project we are creating a system which is capable of tracking the hand gestures and then providing a feedback according to the recognized gesture. The user will be able to use hand movements in the place of a regular mouse inorder to control the operations usually carried out with a mouse. According to the different hand movements, the system will respond and carry out many operations that are available. The gadget mainly consists of a camera and a processor. Here the natural gestures are recognized through analyzing the image frames from the web camera which is focused to the computer monitor. Here, it relies on a user being able to carry out relatively natural motions, movements or gestures that they quickly discover and control the computer application or manipulate the on-screen content.Here first the video stream is received through the camera, which contains the colour markers. Input video is converted to image sequences and each image is processed separately to get the position of the markers. Based on the position and the movement of the colour markers, the gestures are identified. After identifying the gesture, the operation to be done by the gesture will be done.This will be done using terminal commands and other system calls.

CONTENTS

List of Abbreviations

NUI	: *Natural User Interface*
CV	: *Computer Vision*
CSV	: *Comma Separated Values*

List of Figures

List of Tables

CHAPTER 1
INTRODUCTION

1.1 Overview

Here ,the system relies on a user being able to carry out relatively natural motions, movements or gestures that they quickly discover and control the computer application or manipulate the on-screen content. In this project we use the spatial hand gestures for interaction instead of a Mouse or keyboard to control the computers . That is, here we are creating a system which is capable of tracking the hand gestures and then providing a feedback according to the recognized gesture. recognized gestures will invoke the required gestures based on its function.This will be done using terminal commands and other system calls.

1.2 Motivation and Technical Relevance

When we watch movies these days ,all of them seem to have such upscale technology that we could only dream of seeing in our lifetime. So while searching for a topic we became interested in an area called augmented vision and technology.The online searches resulted in the revelation that it was the idea behind one of the TED talks we heard which everyone had deemed unbelievably awesome.So it was the idea that through this project we could also be part of paving the new path for the future in computers.Although this project is not exactly the refined and out-of this world looking system,it will,we hope,be a stepping stone down the path.

Augmented vision and reality is a relatively new field that has seen a lot of activity in the recent years.From smart phones and sports we see on TV ,to wearable computers that will work with just a touch of your hands,we can see augmented reality in many forms.Technically speaking our project is aimed at reducing amount of use of the physical computer screen.It will be simple and easy to use since we are making use of hand gestures.Although the specified hand gestures for certain actions must be learned inorder to use the system,it will be easy to learn.The system is designed so that the users can easily interact with the operating system and manipulate application programs with simple actions .

1.3 Progress of Project

The project begins with with the identification of motivation and technical relevance. The next phase was to identify the hardware and software requirements.After completing this literature survey was begun, which includes many IEEE papers and websites. Next design analysis were done which include module breakup and ended with design of DFD.Then the implementation and the testing were done.

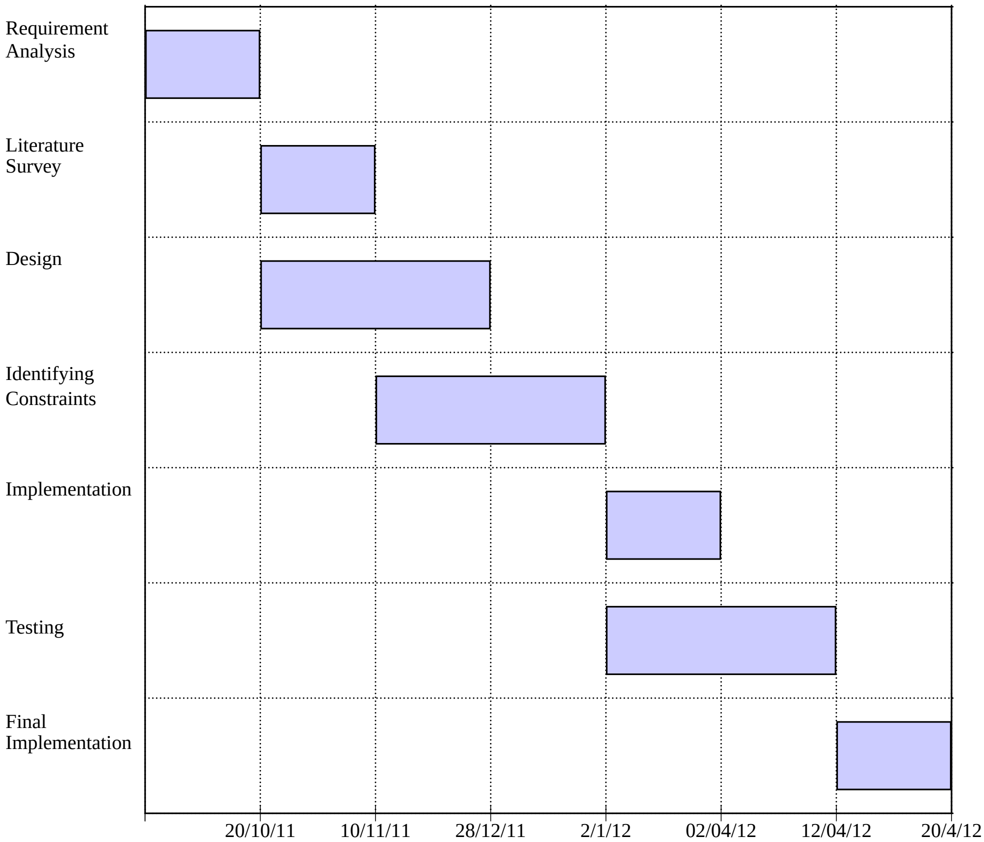

Fig. 1.1: Progress of Project

1.4 Member Role and Responsibilities

After a brief overview of team organization, a table showing the team roles and responsibilities can be shown as below.

Table. 1.1: Team Organization

Name	Role/Responsibility
Anu Pauly	Designer
Renitto Jose	Designer
Shon Joice	Programmer
Sreelakshmy N	Programmer
Tom Jose	Programmer

1.5 Layout

Chapter 2 presents the relevant papers referenced during the initial survey of the project concept.

Chapter 3 presents the details of the proposed system.It also includes the details about the process model used by the system ,which is ,the evolutionary model.

Chapter 4 includes the hardware and software requirements for the project.

Chapter 5 gives an overview of the design and the analysis. It include module breakup, member work effort and module allocations to each member here as per his/her responsibility.The section also presents the general architecture of our project concept. Also design models,that is,the control flow diagrams, flowcharts and block diagram that comprise the design elements of the project, are included.

Chapter 6gives an overview about the implementation details.

Chapter 7 decribes the diffferent testings and a few test cases that were carried out .

Chapter 8 gives the conclusion and the future scope of the project.

CHAPTER 2
LITERATURE SURVEY

This chapter describes the major papers that were reviewed during the literature survey.

2.1 Dynamic Hand Gesture Recognition Using the Skeleton of the Hand

The hand gestures can be divided into two Static hand gestures-represented by a single image of the hand Dynamic hand gestures -represented by a sequence of images, each one corresponding to a hand posture within the gesture . The recognition of a gesture is performed using the fusion of a static and a dynamic recognition technique .The static recognition is used to detect the start and end positions of the hand during the gesture. The dynamic recognition is only performed between the start and stop images.

General process is composed of four main parts :

(i) A pre-processing step to focus on the gesture.

(ii) A feature extraction step where two techniques are used.

The first one, dedicated to static recognition, uses histograms of local orientations in the gesture image (by computing the local gradient on the image). It is based on an algorithm proposed by Freeman and Roth. The orientation histogram will act as the feature set of the gesture.

The second one, used for dynamic gesture recognition, extracts the dynamic signature and is defined using a new original method which gives a compact and efficient representation of the gesture. The dynamic signature is a binary image which represents the superposition of all the skeletons of the hand region for all images (hand postures) within the gesture image sequence. It reflects the motion information of the gesture and also the hand spatial configuration during the gesture motion. Using the proposed dynamic signature, the gesture can be segmented into its constitutive steps such as the preparation and the stroke . Each gesture will be represented by a particular dynamic signature.

(iii) A training step. In order to perform the gesture classification, the gesture alphabet has to be known. A training stage is required. Using a set of samples for each gesture, a training set of data is generated for both the static and the dynamic recognition. Using different representations in space of the same gesture gives invariance to small translations and rotations.

(iv) A classification step. The unknown gesture features will be compared with the features stored in the training data. The best match will lead to the recognition of the unknown gesture.

Static Hand Gesture Recognition

A static hand gesture is represented by a single grey-level image consisting of a particular finger-palm-thumb hand configuration. The static hand gesture recognition is performed by using the local orientation histograms. The Euclidean distance is used as a measure of distance between different orientation histograms. The new gesture orientation histogram is computed and compared with all the histograms from the training set. The minimum Euclidean distance gives the recognized static gesture.

Dynamic Hand Gesture Recognition

A dynamic gesture consists of a particular movement of the hand which encapsulates a particular piece of information The dynamic gestures are represented by a sequence of grey level images. The extraction of the hand region skeleton is performed using a distance transformation-based method. [1]

2.2 Enhancing Hand Gesture Recognition using FCME Model

Previous systems cannot distinguish hand gesture accurately since they use only single model for recognition. Therefore we can obtain efficient hand gesture recognition with its enhanced performance using FCME model .

Proposed method:

It mainly uses multiple local experts obtained via FCME model. The decisions from them are combined with the gating network. Performance is evaluated by comparison with alternative models for hand gesture recognition . Improved gesture recognition performance is obtained on similar hand gesture recognition. [2]

Mixtures-of-Experts

It uses s divide-and-conquer principle and then reducing the complexity of the recognition process. It mainly consist of a set of experts and a gating network. Each expert deals with different features from a different perspective. The complexity of the problem can be reduced by dividing all hand gestures into some subgroups.

2.3 Hand Gesture Recognition in Natural State Based on Rotation Invariance and OpenCV Realization

The advantage of hand gesture recognition method based on computer vision is that the hands can be in natural state, and make the human interactive with computer naturally. The natural state hand gesture recognition algorithm can detect human hand in real-time from given images, and the given images could be acquitted from natural environment. In order to obtain the detection results from real time hand images, we should find steady-going hand gesture features from the images, and the features does not change when the acquisition conditions of images are changed such as the illumination, image size, and the rotation angles. The most important feature of this approach to recognize hand gestures is that it has rotation invariance and the real-time hand gesture recognition can get rid of the interference in rotate images. [3]

The Fundamentals of Rotation Invariance Hand Recognition

The hand gesture recognition method mainly use the monocular vision based on exterior features. Firstly, adjust the brightness of the image sequences that acquired from the camera. Use skin color model to extracting skin pixels and locating hand gestures region, and segment the hand area from complex background with the help of shin color maintenance. Then extract and analyze the pixels feature which has been acquired from the hand area, and extract its characteristic parameter to create hand gesture models with the virtue rotation invariance. Lastly, the recognition results could be obtained by using Gaussian model to regulate related parameters.

Gesture Segmentation in Complex Background

The basic goal of gesture segmentation is creating the color model of the hand and extracting skin color information in the appropriate color space. The hand gesture segmentation can dislodge most of background area, while the illumination factor or other factors would affect the segmentation, and some valid information would be dropped. The fact is that although there may be a far cry among different peoples skin color but the chromaticity difference is far less than skin color difference. That is to say, the chromaticity among different peoples skin color are usually similar to each others, and only the brightness difference is the principal factor. Therefore, the clustering features of skin color information can be used segment the skin color rapidly.

2.4 A Robust and Accurate 3D Hand Posture Estimation Method for Interactive Systems

Existing hand gesture recognition systems estimate hands 3 model based on image features such contour or skin texture. But it is difficult to estimate the wrist rotation because the contour and the texture data do not have enough information to distinguish hands sides. To solve this problem a new 3D hand posture estimation system that uses data of nail positions is used. Nail positions are an important factor to recognize hand sides using nail positions .It becomes possible to detect whether the camera is facing palm or dorsum .In addition, nail areas can be robustly extracted from a skin area by a simple image processing technique. The system uses a database consisting of data sets of the hands contour , the nail positions and the finger joint angles. To estimate the hand posture , the system first extracts the hands contour and the nail positions from the captured image. Then searches for a similar data set from the database. System then outputs the finger joint angles of the searched data set. [4]

2.5 Vision Based Hand Gesture Recognition

With the development of ubiquitous computing, current user interaction approaches with keyboard, mouse and pen are not sufficient. Gestures are a powerful means of communication among humans.Because direct use of hands as an input device is an attractive method for providing natural Human Computer Interaction. A natural interaction between humans and computing devices can be achieved by using hand gestures for communication between them.But the key problem is how the gestures are understood by the computer.The approaches can be divided into data glove method and vision based method. Here we use vision based technique because it requires only camera rather than sensor devices in data glove method. [5]

CHAPTER 3

PROPOSED SYSTEM

This chapter presents the proposed system in detail as well as the process model that is used in the project.

3.1 The NUI system

Natural user interface, or NUI, is an emerging paradigm shift in man machine interaction of computer interfaces to refer to a user interface that is effectively invisible, or becomes invisible with successive learned interactions, to its users. Here we use the natural interaction .That is, most computer interfaces use artificial control devices whose operation has to be learned. A NUI relies on a user being able to carry out relatively natural motions, movements or gestures that they quickly discover control the computer application or manipulate the on-screen content. It refers to a system in which the physical surroundings of a person are mixed with real time computer generated information creating an enhanced perception of surrounding environment. In this project we are creating a system which is capable of tracking the hand gestures and then providing a feedback according to the recognized gesture.

The gadget mainly consists of a camera and a processor. Here the natural gestures are recognized through analyzing the image frames from the web camera which is focused to the computer monitor.

This process mainly involves a series of steps. Here first the video stream is received through the camera, which contains the colour markers. Input video is converted to image sequences and each image is processed separately to get the position of the markers. Based on the position and the movement of the colour markers, the gestures are identified. After identifying the gesture, the operation to be done by the gesture will be done.This will be done using terminal commands and other system calls. [6] [7]

Here we initially grab the frames from the input video. Then the frames are converted to the HSV colour format from the RGB format. Then the image is binarized according to the minimum and maximum HSV value for the required colour. Ie, if the value comes within the particular range ,it is made white and otherwise black. Thus with that we identify the colour band. Then find its contour and the smallest bounding rectangle and hence find its median to locate the coordinates. With the identified coordinates, check the condition for each of the gesture and then identify the right gesture. After identifying the gesture, the operation to be done by the gesture will be done.This will be done using terminal commands and other system calls.

This process is illustrated in the following fig 3.1.

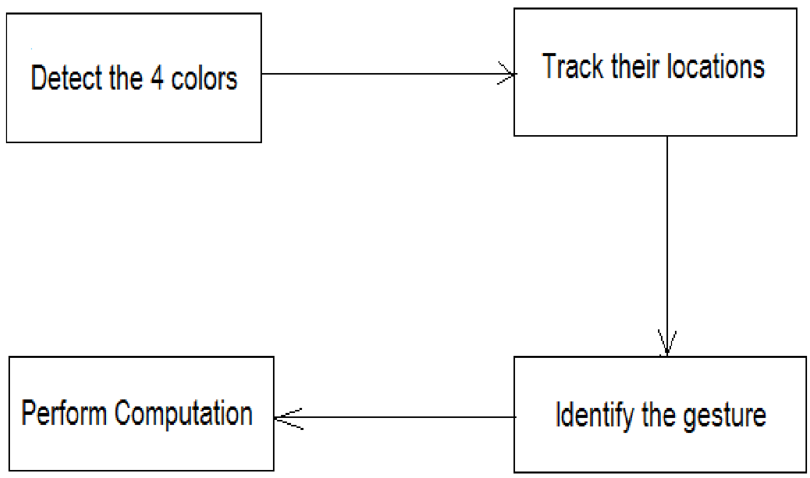

Fig. 3.1: The NUI System

Thus the project aims at bringing the real world and virtual world together. The user will be able to use hand movements in the place of a regular mouse inorder to control the operations usually carried out with a mouse. According to the different hand movements, the system will respond and carry out many operations that are available. Integration of reality and the computer system is apparent in this project where a computer can be operated in a simplified manner using our hand.

3.2 Process Model

Our project uses evolutionary process model. A process model that views development as a series of hills,each representing a separate loop of the spiral model.This model has been chosen, because,if any failures occur, they are accepted and the project is redone from where the failure might have occured and not from the start.This model incorporates efficiency of including the user feedback and also additional requirements during the development of the project.The Evolutionary models take the concept of evolution into engineering paradigm.Therefore evolutionary models are iterative. They are built in a manner that enables the software engineers to develop increasingly more complex versions of the software.

This Incremental Model combines the elements of the Linear Sequential Model with the iterative philosophy of prototyping.When the incremental model is used the first increment is the core product.The subsequent iterations are the supporting functionalities or the add on features that a consumer would like to see.

CHAPTER 4
SYSTEM REQUIREMENT SPECIFICATION

4.1 Hardware Specifiction

- Processor : Dual Core Processor with a clock speed of 2.40Ghz or above
- RAM: 2 GB
- Hard disk : 80GB
- Webcam : Should be capable of recording 640 * 480 at 30fps.

4.1.1 Hardware Description

The major input unit needed for this project is a web cam, which is used to capture live view of the user gestures. A good quality camera with sufficient light sensitivity can be used for the purpose. Whatever be the resolution of the camera, OpenCV with Linux can capture only frames at VGA resolution, putting some restrictions on the quality of frames captured.

4.2 Software Specification

- Operating system : Linux
- Language Used : Python
- Front End : Qt Designer 4
- Libraries used : OpenCV, Xdotool
- Supportinf files : CSV

4.2.1 Python

Python is an interpreted, general purpose high level programming language whose design philosophy emphasizes code readability. Python aims to combine remarkable power with very clear syntax , and its standard library is very large and comprehensive .Its use of identification for a block delimiters is unique among popular programming languages.

Python supports multiple programming paradigms, primarily but not limited to object oriented, imperative and to a lesser extent, functional programming styles. It features a fully dynamic type system and automatic memory management. Python is often used as scripting language, but is also used in wide range of non scripting contexts.

The reference implementation of python is free and open source software and has a community based development model, as do all or nearly all of its alternative implementations. [8]

Python interpreters are available for many operating systems, and Python programs can be packaged into stand alone executable code for many systems using various tools.

4.2.2 OpenCV

OpenCV is a computer vision library originally developed by Intel. It is free for use under the open source BSD license. The library is cross platform. It focuses mainly on real time image processing. If the library finds Intels Integrated performance Primitives on the system, it will use these commercial optimized routines to accelerate it. [9]

OpenCV open Source Computer Vision is a library of programming functions mainly aimed at real time computer vision.

Example applications of the openCV library are Human computer Interaction (HCI). Object identification , Segmentation and Recognition ; Face Recognition ; gesture Recognition ; Motion Tracking , Motion understanding ; Structure From motion ; Stereo and Multi camera Calibration and Depth Computation. [10]

4.2.3 Xdotool

Xdotool lets us simulate keyboard input and mouse activity, move and resize windows , etc. It does this using X11s XTEST extension and other Xlib functions. Additionally you can search for windows and move, resize,hide and modify window properties like the title. If your manager supports it, you can use Xdotool to switch desktops,move windows between desktop, and change the number of desktops.

CHAPTER 5
DESIGN & ANALYSIS

5.1 System Analysis

The system is analysed by dividing it into four separate modules.Major features that are to be provided in each module is decided.This section presents an analysis of the project in terms of the module elaboration and effort distribution.

5.1.1 Module Breakup

This section presents the module break-up.The main modules of the system are the Settings module and the Gesture based Interaction module.

Table. 5.1: Module Description

Module	**Description**
Settings	Store settings & locate colour markers
Gesture based interaction	Check for gesture & execute event

Settings module

The main module reads the user settings stored in a csv file, captures the frames,locates the colour markers, displays the captured frame and also invokes all the other sub modules. It invokes the modules for taking snap shot, invoking the image viewer, interfacing mouse or keyboard , launching explorer,optical drive eject and checking interactive gestures.

Gesture based interaction Module

All the functions like taking snap shot, invoking the image viewer, interfacing mouse or keyboard , launching explorer,optical drive eject and checking interactive gestures are done here.Most of these sub modules contain sub-sub modules for extended functionalities and improved intuitiveness.Here some condition that is defined for identifying a particular gesture is checked and then corresponding event is executed if the matching is right.

5.1.2 Member Effort

This section presents each member's effort in the team.

Table. 5.2: Module Allocation

#	Task	Estimated Effort	Start Date	End Date	Person
1	Settings	(95)	(1/02/2012)	(8/03/2012)	All members
2	Gesture based interaction	(130)	(9/03/2012)	(15/04/2012)	All members

5.2 System Design

System design was done to identify the major features to be provided in the system and its sub modules. In this project we are create a system which is capable of tracking the hand gestures and then providing a feedback according to the recognized gesture. The overall design of the project is as follows.The gadget mainly consists of a camera and a processor. Here the natural gestures are recognized through analyzing the image frames from the web camera which is focused to the computer monitor. [11]

The process can be briefed as follows- The video stream is captured by using the PC webcam. The video stream input is converted into an image sequence. Each frame is traversed pixel wise to obtain the positions of the required colours. The pixels which correspond to the images in which the colours exeed the preset value is marked. Based on the colours being detected and their movements, the gestures are identified and the required operation is done on the computer.

5.2.1 System Architecture

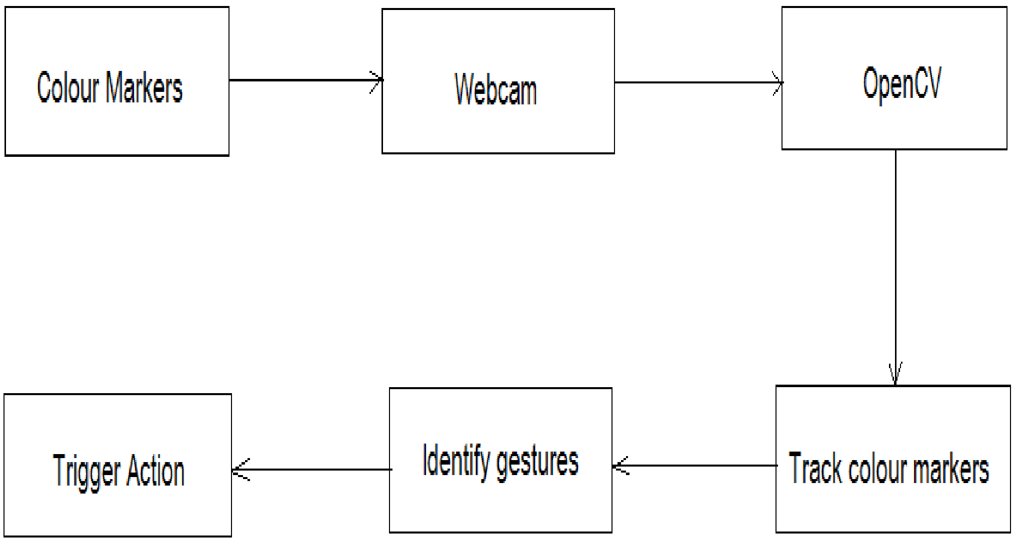

Fig. 5.1: System Architecture

Our project architecture starts with the capturing of video frames from a webcam and processin it frame by frame .Each frame is checked for the existence and locations of the four colour markers, ie, red, yellow,blue and green, making use of the built in functions provided by the OpenCV library. The position of the colour markers are tracked continuously and based on their movements and relative positions, various gestures are identified and corresponding actions are triggered.

5.2.2 Control Flow Diagrams

A Control Flow Diagram is a diagram to describe the control flow of a business process , process or a program. Here the main module reads the user settings stored in a csv file, captures the frames,locates the colour markers, displays the captured frame and also invokes all the other sub modules. It invokes the modules for taking snap shot, invoking the image viewer, interfacing mouse/keyboard , launching explorer,optical drive eject and checking interactive gestures.

Most of these sub modules contain sub-sub modules for extended functionalities and improved intuitiveness.

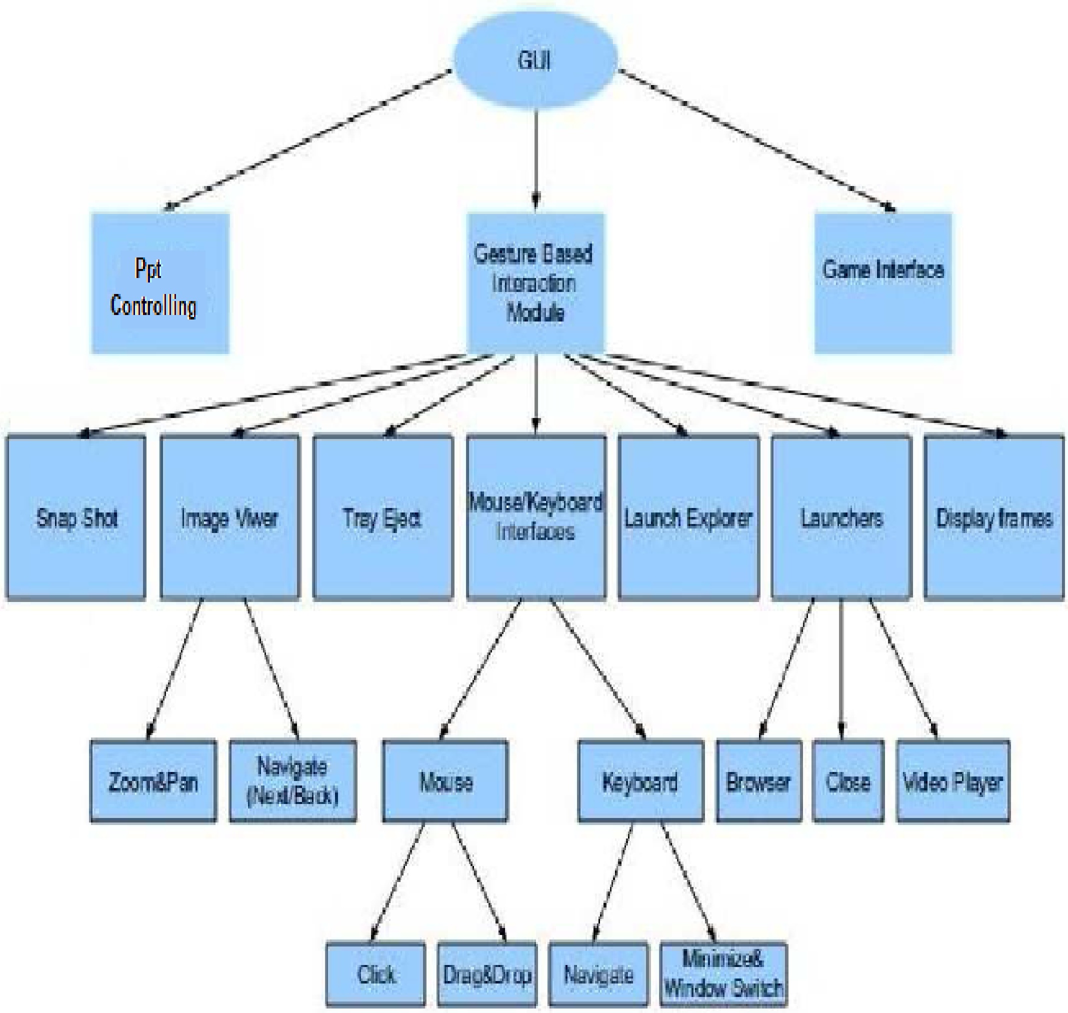

Fig. 5.2: Control Flow Diagram

5.2.3 Flow Chart

Here we begin by start capturing the video. Then we try to locate the coordinates of the colour markers. Then its position or movement is identified. Then we check whether it is the gesture for taking a snap. If yes, then we save the current frame. If it is not, then we check the gesture for launching the image viewer.If it is, then start the image viewer.Then we check for the gestures for navigating front or back, zoom in or zoom out, etc. If it matches,then the corresponding action will be carried out.If the gesture is not for launching image viewer , then check whether it is the gesture for mouse/keyboard interface. If it is, then perform the corresponding actions according to the gestures.Else check whether it was the gesture for launching Explorer. If it is, then load the specified explorer.If the gesture was not for that,then check whether it is for the CD eject and if it is, then have the system call for that.If it was not even that,then check for the interactive gestures and perform the required actions.

The flow chart is represented by the following figures.

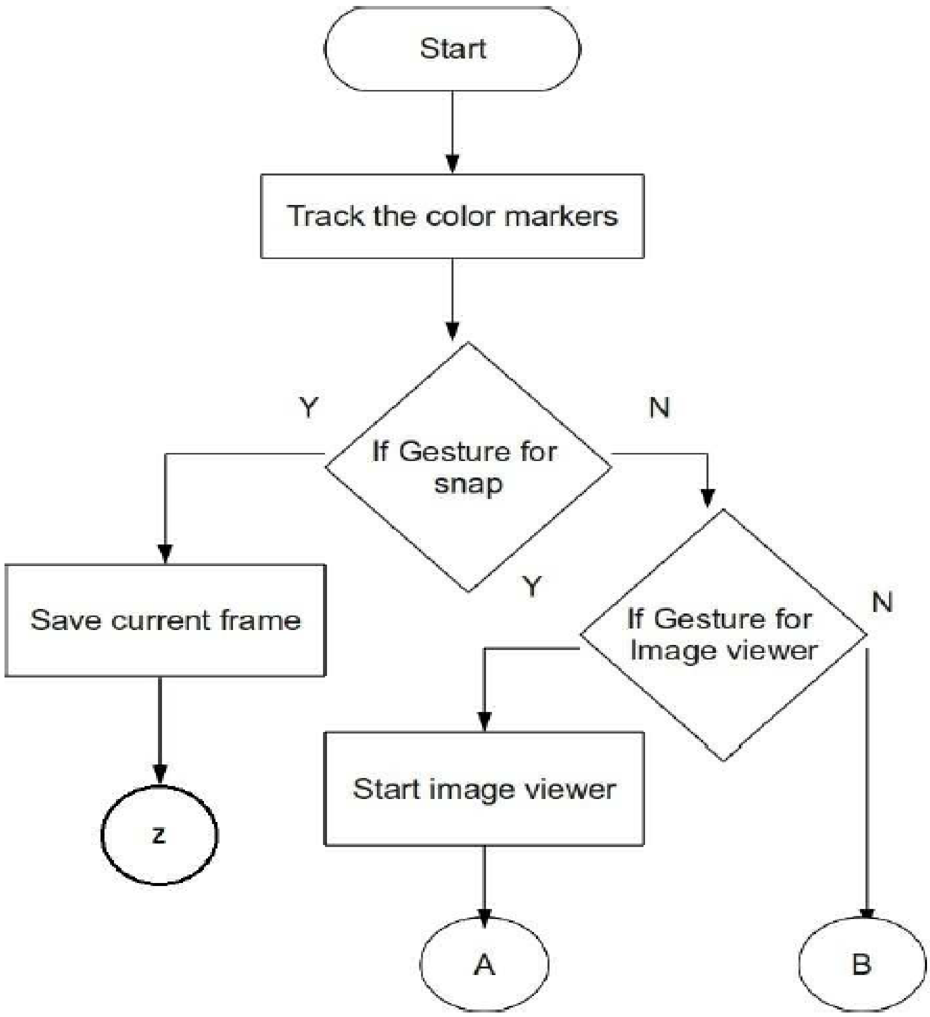

Fig. 5.3: FlowChart1

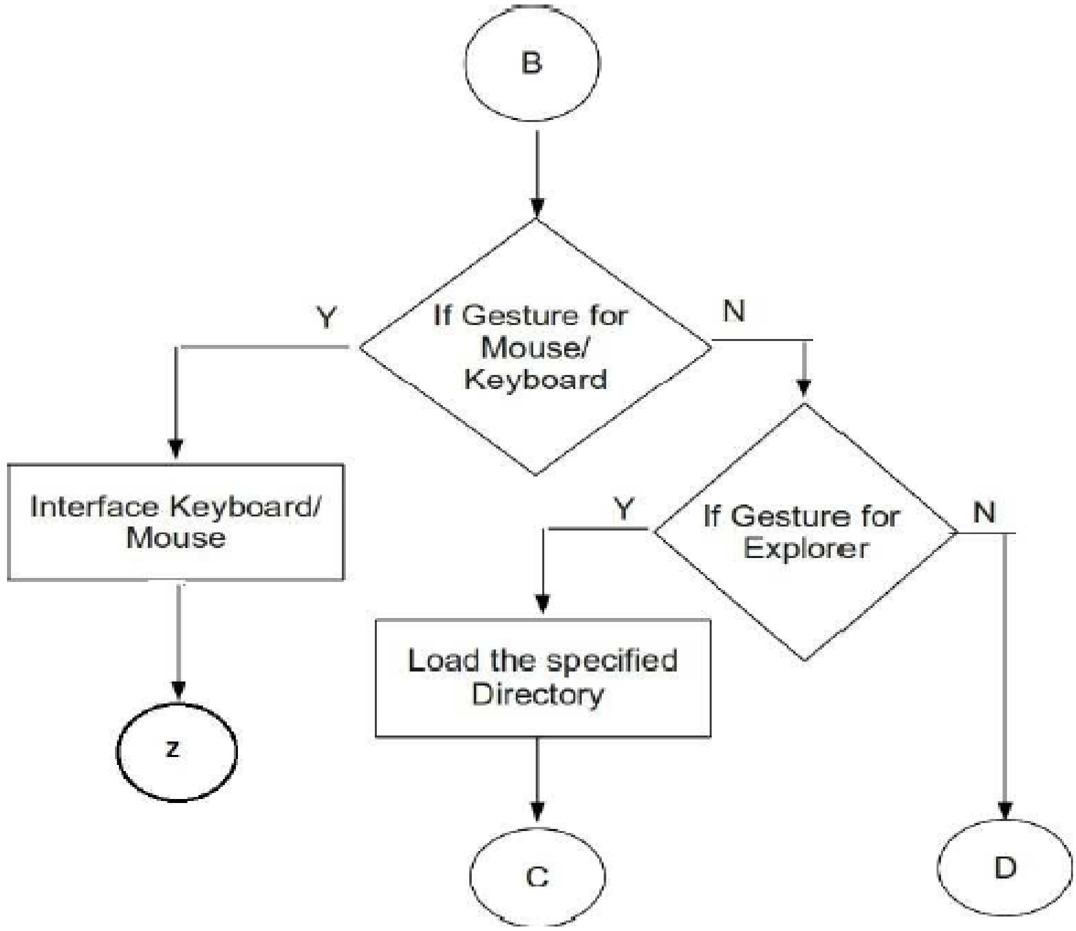

Fig. 5.4: FlowChart2

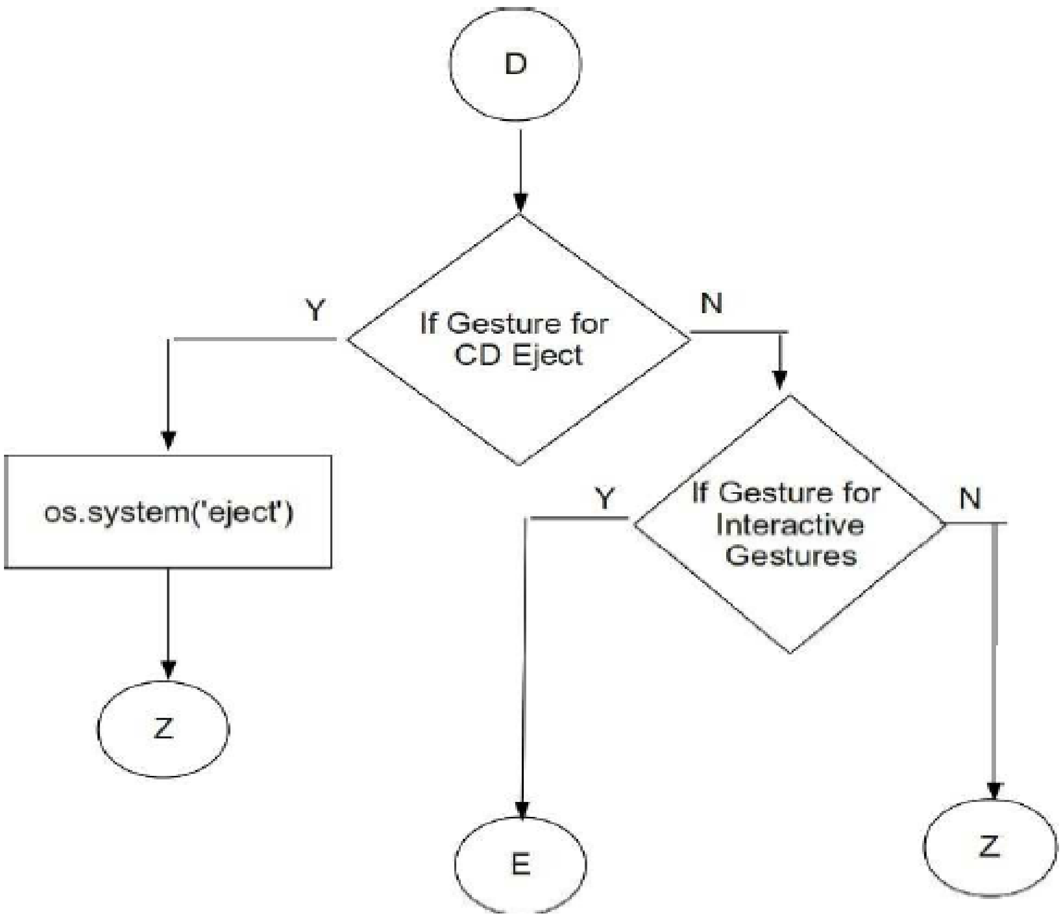

Fig. 5.5: FlowChart3

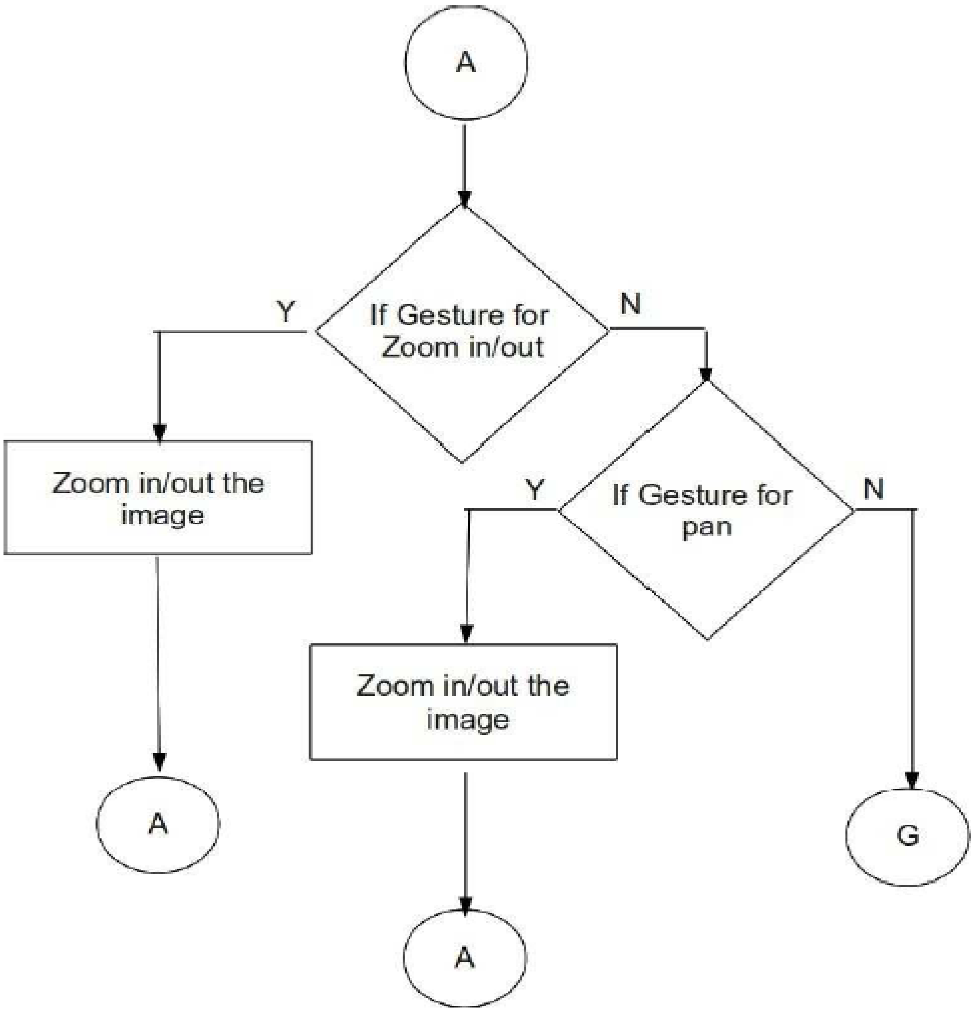

Fig. 5.6: FlowChart4

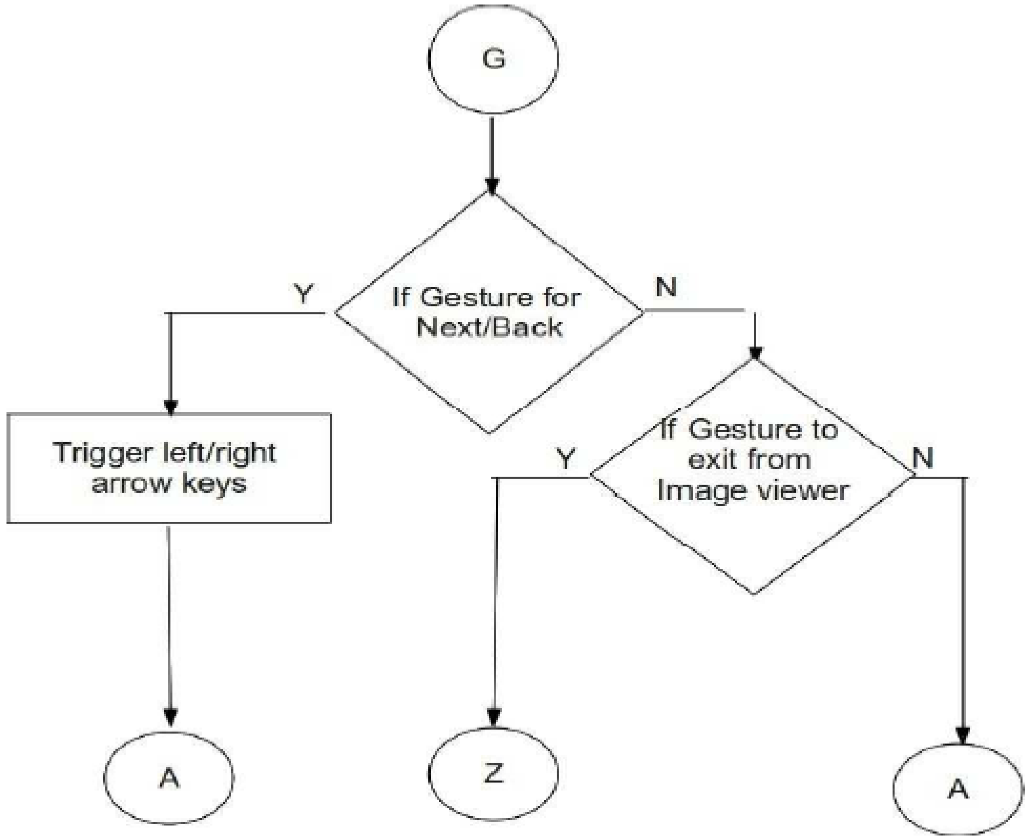

Fig. 5.7: FlowChart5

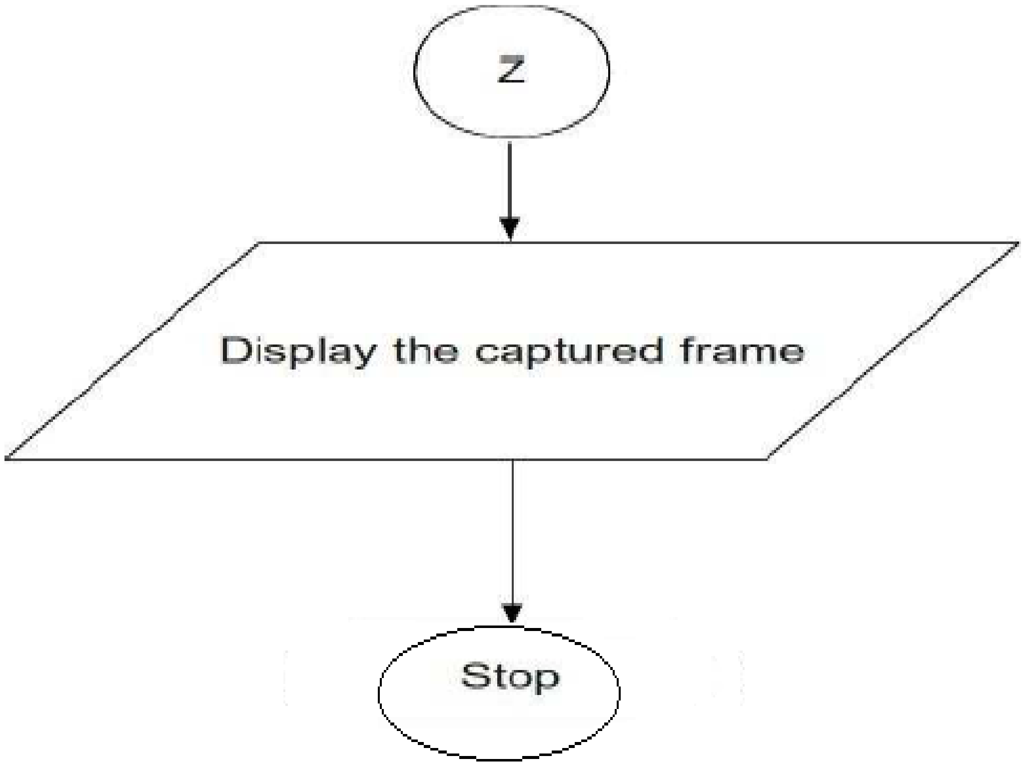

Fig. 5.8: FlowChart6

CHAPTER 6
IMPLEMENTATION

6.1 Module wise description

This section, presents the description of each of the modules.The different modules of the system are Settings module and the Gesture based interaction Module.

Settings module

The main module reads the user settings stored in a csv file, captures the frames,locates the colour markers, displays the captured frame and also invokes all the other sub modules. It invokes the modules for taking snap shot, invoking the image viewer, interfacing mouse or keyboard , launching explorer,optical drive eject and checking interactive gestures.

Gesture based interaction Module

All the functions like taking snap shot, invoking the image viewer, interfacing mouse or keyboard , launching explorer,optical drive eject and checking interactive gestures are done here.Most of these sub modules contain sub-sub modules for extended functionalities and improved intuitiveness.Here some condition that is defined for identifying a particular gesture is checked and then corresponding event is executed if the matching is right.

Implementation Details

This section gives the implementation details of the project. It mainly involves two kinds of gestures - the Static gestures and the Interactive gestures.

Static Gestures

- Creating a Photo Frame with the hands takes a photo using the webcam.
- Creating an Eye shape with the left hand invokes the Image Viewer to browse the images taken by the program. Based on the gestures, the images can be browsed easily.
- Optical Drive can be ejected by creating a Drive Shape using both hands.

Interactive Gestures

- Moving the Index finger moves the mouse on screen. When the thumb is shown a double click is triggered.
- There is also a Drag n Drop feature to move files,text,windows around.
- Drawing an X closes the currently active application.
- Drawing a C launches the user defined Web Browser.
- Drawing a V launches the user defined Media Player.
- Zoom & Pan photos using Pinch gesture.
- Minimizing the active window.
- Launching the user specified location in the Explorer.
- Switching Windows.
- Navigating through Files or Folders using Mouse & Keyboard gestures.
- Launch presentation and navigate through the slides.

- Play the game, Angry Birds.

6.2 Implementation Areas

This project can be implemented on any system with a web cam to get an intuitive & interactive interface. Its especially useful for interactive gaming & multimedia systems.

6.3 Screen Shots

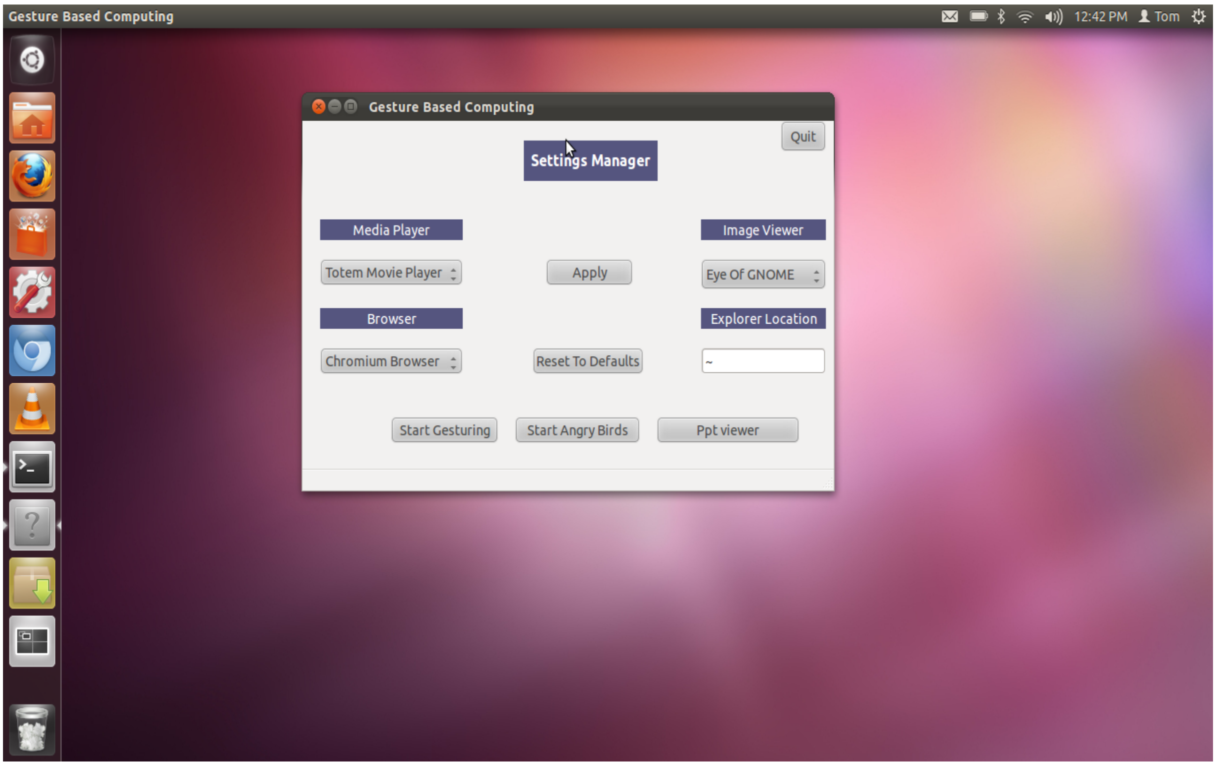

Fig. 6.1: Interface

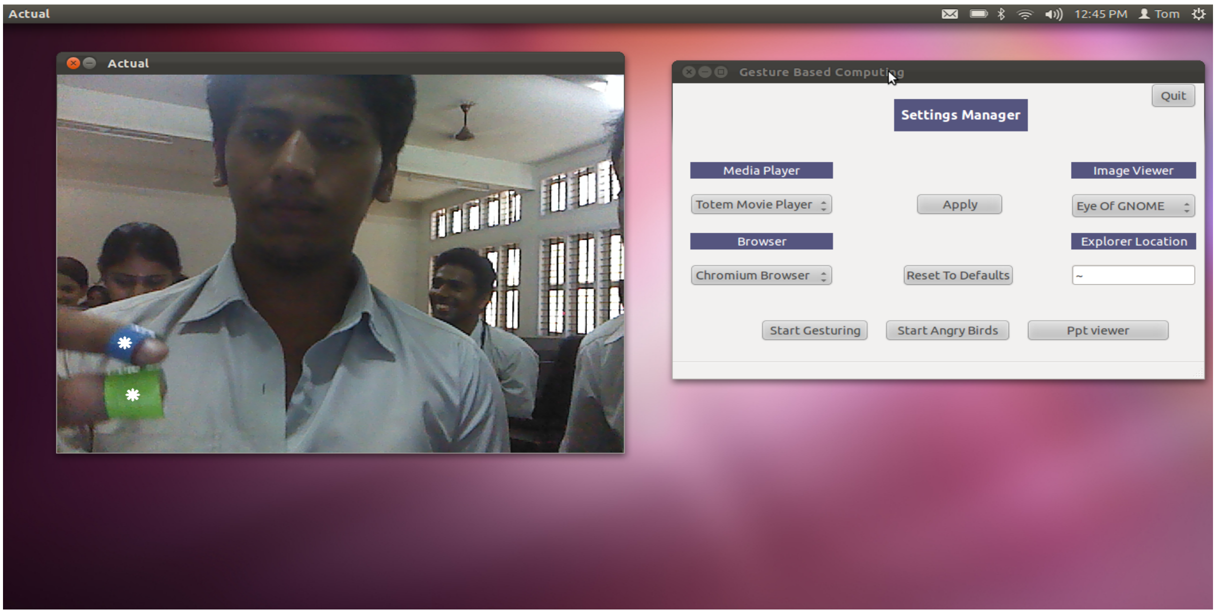

Fig. 6.2: Gesture for launching Image viewer

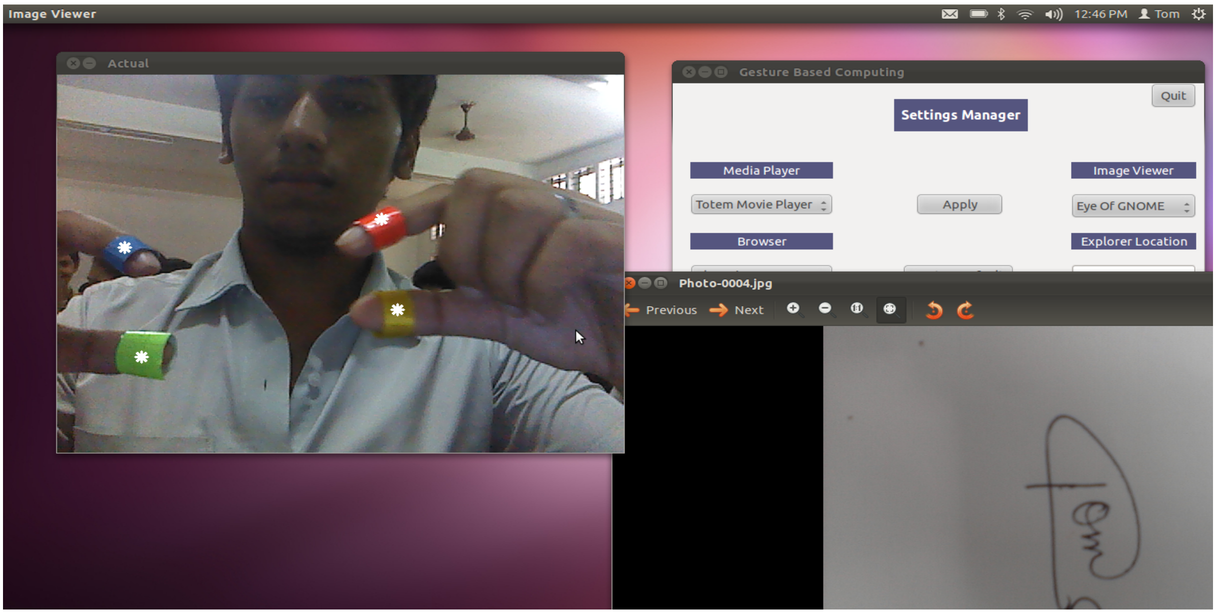

Fig. 6.3: Zooming gesture

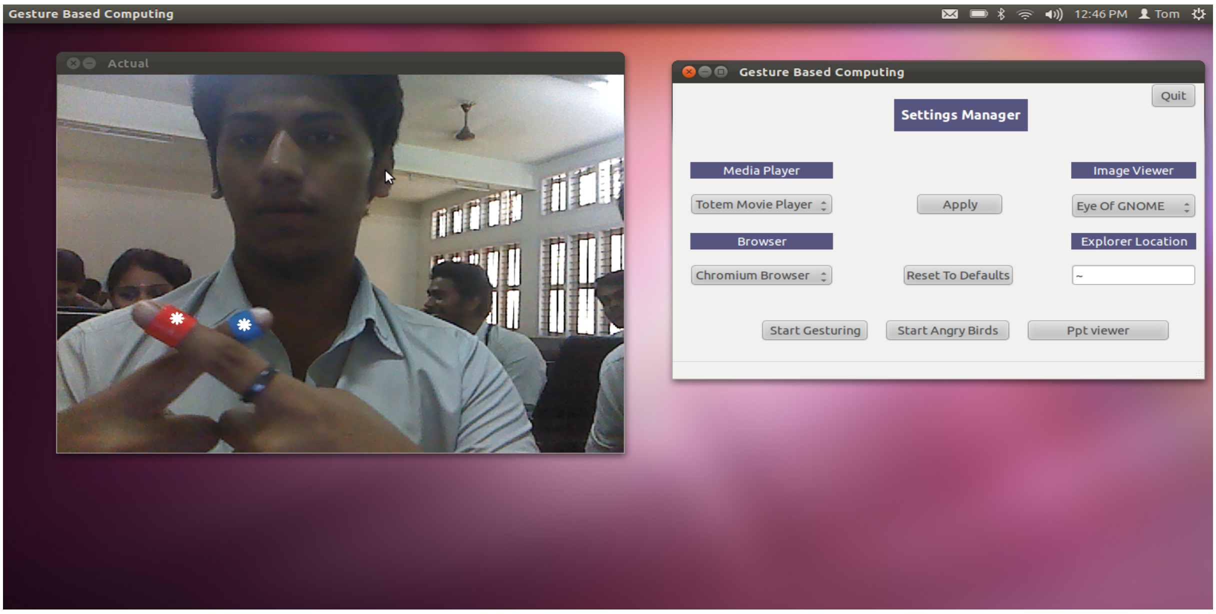

Fig. 6.4: Gesture for Closing

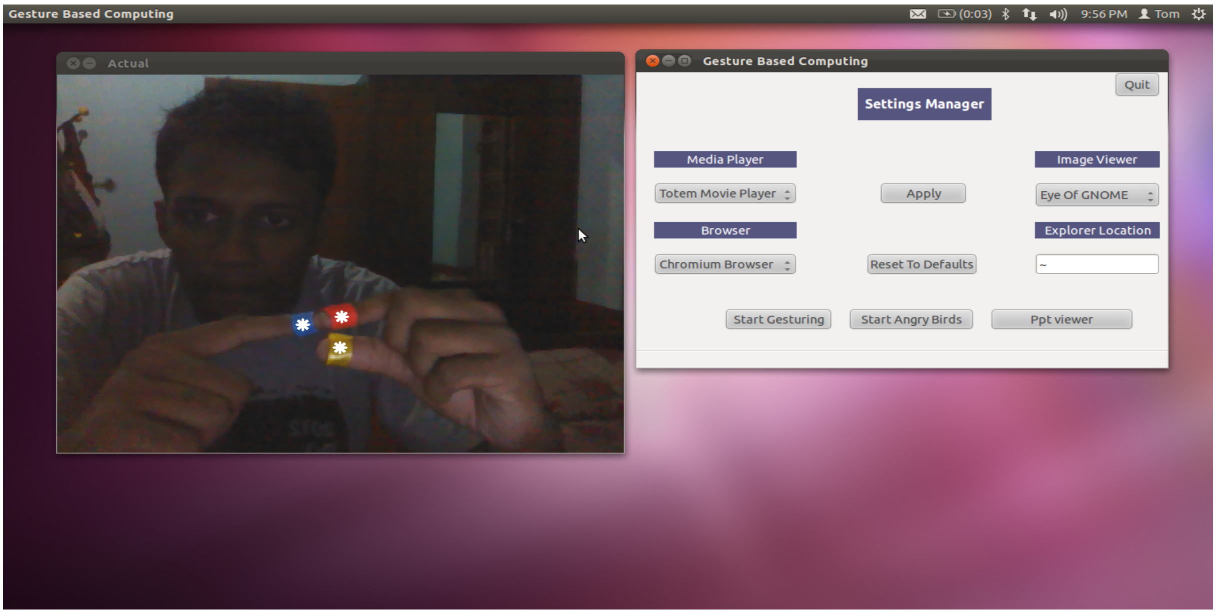

Fig. 6.5: Gesture for activating mouse

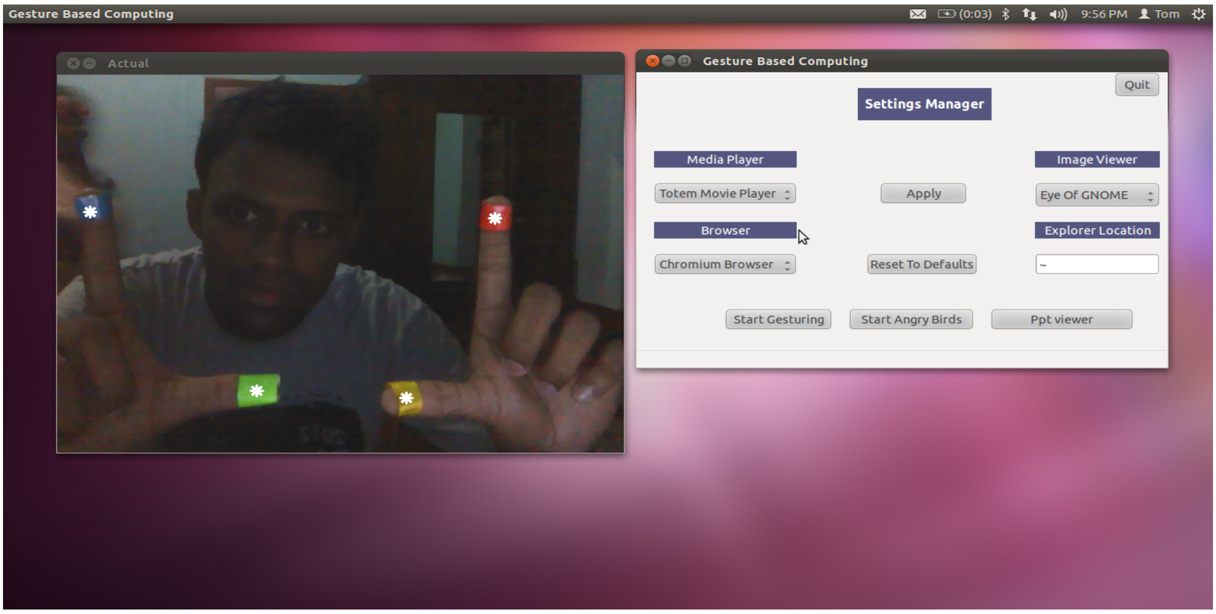

Fig. 6.6: Gesture for window toggling

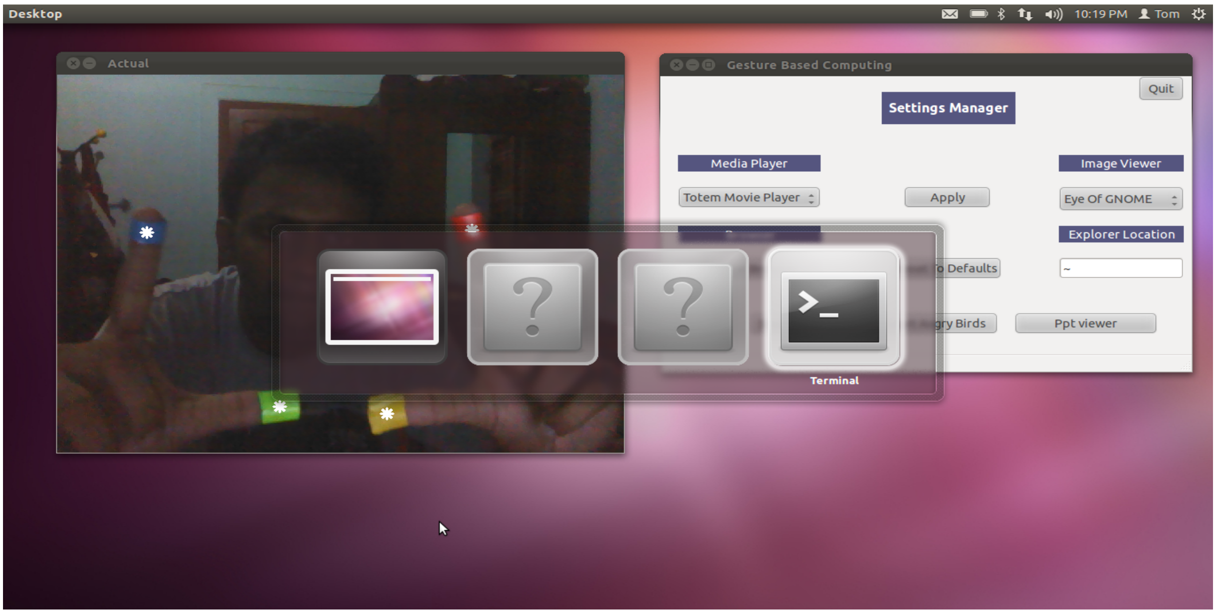

Fig. 6.7: Windows

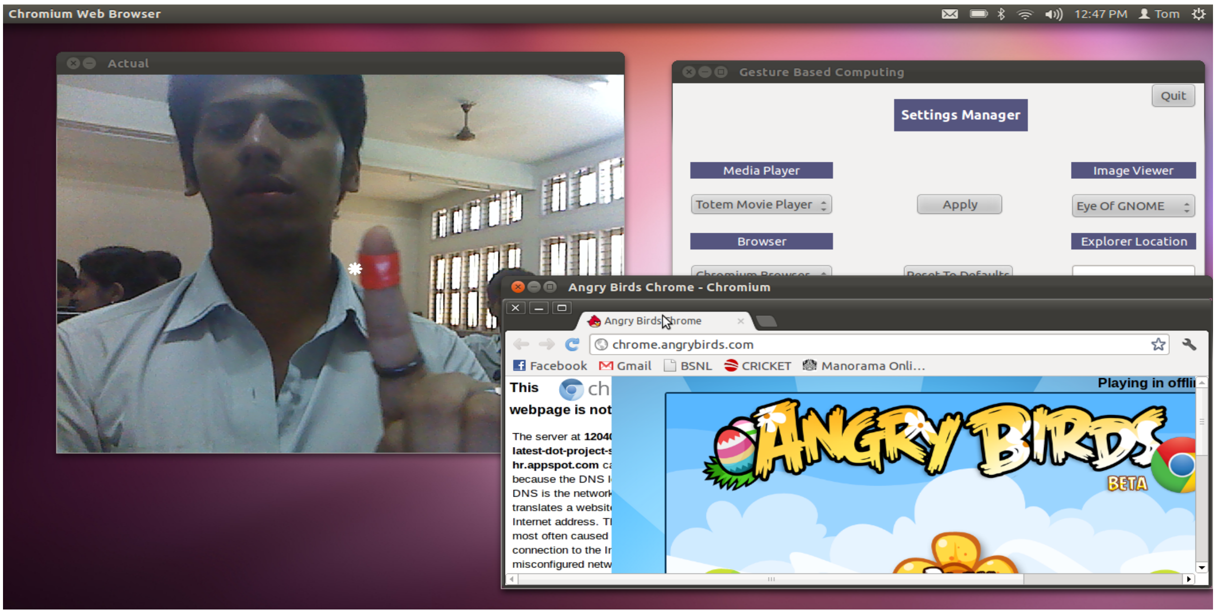

Fig. 6.8: Playing Angry birds

Fig. 6.9: Gesture for starting Ppt

6.4 Limitations

- A major limitation in this system is problems that can occur due to poor lighting conditions.
- The human arm held in an unsupported horizontal position rapidly becomes fatigued and painful.
- Presence of other materials with same colour as the markers may cause lack in efficiency.

CHAPTER 7
TESTING

7.1 Introduction

System testing of software or hardware is testing conducted on a complete, integrated system to evaluate the systems compliance with its specific requirements. As a rule ,system testing takes, as its input, all of the integrated software components that have successfully passed integration testing and also the software system itself integrated with any applicable hardware systems. The purpose of integration testing is to detect any inconsistencies between the software units that integrated together or between any of the assemblages and the hardware.

7.2 Unit Testing

Unit testing refers to tests that verify the functionality of a specific set of code usually at the function level.In an object oriented environment, this is usually at the class level, and the minimum unit tests include the constructors and destructors. Unit testing alone cannot verify the functionality of a piece of software, but rather is used to assure that the building blocks the software use work independently of each other.

Each module in the software , ie, the settings module and the gesture based interaction module were tested individually and the results were noted.The modules were tested by executing them individually.

7.3 Integration Testing

Integration testing is any type of software testing that seeks to verify the interfaces between components against a software design. Software components may be integrated in an iterative way or all together.

Integration testing works to expose defects in the interfaces and interaction between integrated components.Progressively larger groups of tested software components corresponding to the elements of the architectural design are integrated and tested until the software works as a system.

We tested the integrated software while adding each module to the interface. The small defects were corrected.

7.4 System testing

System testing tests a completely integrated system to verify that it meets its requirements. The system meets its requirements.

7.5 Test Cases

Testing the logic : Checked whether colours and gestures were detected from the input. The positive test case was that the colours were detected and the gestures were identified. But the negative test case was that the output varies according to the lighting condition and the background condition.

The final testing : The different gestures were identified successfully and they were executed on the system. The positive test case was that the different gestures were recognized and the operation was performed according to the computer. The negative test case was that due to poor lighting condition some gestures were not recognized.

7.6 Maintenance

Maintenance includes upgrades, fault correction and performance improvements.Faults may be the results of either design errors and or implementation errors.Design errors represent a very significant proportion of the catastrophic failures that occur. Design errors often corresponds to situations which are hard to test, or easy to overlook. Implementation errors are also called programming errors,these are when the actual software or hardware do not follow the intended design. We have made earnest efforts to correct all possible errors in our system.However we have borne in mind the fact that the software environment is constantly changing and have coded accordingly.Also we have ensured that even if maintenance is required it will be cost effective.

CHAPTER 8
CONCLUSION

Here the project present the method to use spatial hand gestures for interaction, instead of a Mouse or keyboard, to control the computers. We will be able to control the system using a predefined set of hand gestures.Although the specified hand gestures for certain actions need to be learned inorder to use this system,it will be easy to learn.The system can be designed in a way such that the users can easily interact with the operating system and manipulate application programs with simple actions . This means that, by the completion of this project, the user will be able to interact with a regular personal computer with the use of their hands.

Future Work

This software can be integrated with the OS to provide improved inbuilt support for gestures to the users.

Gesture-based computing has strong potential in education, both for learning, as students will be able to interact with ideas and information in new ways, and for teaching, as faculty explore new ways to communicate ideas. It can also be used for Interactive Gaming & Multimedia applications. It can also be used to control the secondary devices in cars & other high tech machines.

REFERENCES

[1] B. Ionescu and D. Coquin, "Dynamic hand gesture recognition using the skeleton of the hand," *EURASIP Journal on Applied Signal Processing*, 2005.

[2] J. V. Yoon and J. K. Min, "Enhancing hand gesture recognition using fuzzy clustering based mixture of experts model."

[3] B. Zhang, R. Yun, and H. Qiu, "Hand gesture recognition in natural state based on rotation invariance and opencv realization."

[4] E. Tamaki, "A robust and accurate 3d hand posture estimation method for interactive systems."

[5] P. Garg and N. Aggarwal, "Vision based hand gesture recognition."

[6] Gesture-based computing. [Online]. Available: http://wp.nmc.org/horizon2011/sections/gesture-based-computing/

[7] M. Porta, "Vision-based user interfaces: methods and applications," *International Journal of Human-Computer Studies*, 2002.

[8] A. Downey, J. Elkner, and C. Meyers, "How to think like a computer scientist: Learning with python," *Green Tea Press*, 2008.

[9] Tracking colored objects in opencv. [Online]. Available: http://www.aishack.in/2010/07/tracking-colored-objects-in-opencv/

[10] G. Bradski and A. Kaehler, "Learning opencv: Computer vision with the opencv library," *O'Reilly Media*, 2008.

[11] A. Sepehri, Y. Yacoob, and L. S. Davis, "Employing the hand as an interface device," *Journal of Multimedia*.

Appendix

Finding coordinates

```
def CalcPos(min_color,max_color):

        global frame

        cv.Smooth(frame,b_frame, cv.CV_GAUSSIAN, 9, 9)
    cv.CvtColor(b_frame, imgHSV, cv.CV_BGR2HSV)
    cv.InRangeS(imgHSV,min_color,max_color,imgMask)

        cv.Erode(imgMask,imgMask, None ,4)
        cv.Dilate(imgMask,imgMask, None ,4)
        contour = cv.FindContours(imgMask, storage, cv.CV_RETR_CCOMP,
            cv.CV_CHAIN_APPROX_SIMPLE)
    points = []
        c1_x,c1_y,c2_x,c2_y=0,0,0,0
        count_pix=0
        x,y=1,1


        while contour:
                bound_rect = cv.BoundingRect(list(contour))
                if contour:
                        area= abs(cv.ContourArea(contour))

                        if area : #in range(10,3000):

                                c1_x+=bound_rect[0]
                                c1_y+=bound_rect[1]
                                c2_x+=bound_rect[2]+bound_rect[0]
                                c2_y+=bound_rect[3]+bound_rect[1]
                                count_pix+=1

                contour = contour.h_next()



        if count_pix :
```

```
            c1_x/=count_pix
        c1_y/=count_pix
        c2_x/=count_pix
            c2_y/=count_pix
                x=(c1_x+c2_x)/2
                y=(c1_y+c2_y)/2


            pt1,pt2=(c1_x,c1_y),(c2_x,c2_y)
            return x,y
        return 0,0


def GetCoords():

        global frame,xy,yy,xr,yr,xg,yg,xb,yb
        frame=cv.QueryFrame(capture)
        xy,yy=CalcPos(ymin,ymax) #assume no color detection 1st
        xg,yg=CalcPos(gmin,gmax)
        xb,yb=CalcPos(bmin,bmax)
        xr,yr=CalcPos(rmin,rmax)
        MarkColor(xr,yr)
        MarkColor(xy,yy)
        MarkColor(xb,yb)
        MarkColor(xg,yg)
        return xy,yy,xr,yr,xg,yg,xb,yb

def MarkColor(x,y):
        global frame
        if x and y:
                cv.Line(frame, (x,y-7), (x,y+7), cv.CV_RGB(255,255,255),2)
                cv.Line(frame, (x-7,y), (x+7,y), cv.CV_RGB(255,255,255),2)
                cv.Line(frame, (x+5,y+5), (x-5,y-5), cv.CV_RGB(255,255,255),2)
                cv.Line(frame, (x-5,y+5), (x+5,y-5), cv.CV_RGB(255,255,255),2)
```

Checking gesture for snap shot

```
snap_gstr=0
rd=csv.reader(open('snap.csv','rb'),delimiter=' ',quotechar='|')#read snap no
```

```
    from file
for row in rd:
        snap_count=int(''.join(row))
snap_name="snap"


def take_snap(xr,yr,xy,yy,xg,yg,xb,yb):

        global snap_gstr,snap_count,snap_name

        print "found"
        snap_gstr=1

        cv.WaitKey(1500) #1100

        while (xb or xr or xg or xy):

                xy,yy,xr,yr,xg,yg,xb,yb=GetCoords()

        if snap_gstr:
                frame=cv.QueryFrame(capture)
                cv.SaveImage(('/home/vishwakarma/snapshot/'+snap_name+str(snap_count)+".
                    frame)
                cv.DestroyWindow("Actual")
                cv.MoveWindow("Snap Shot",60,60)

                cv.ShowImage("Snap Shot",frame)
                cv.MoveWindow("Snap Shot",60,60)

                snap_count+=1
                wr=csv.writer(open('snap.csv','wb'), delimiter=' ',
                    quotechar='|',quoting=csv.QUOTE_MINIMAL)
                wr.writerow(str(snap_count))
                snap_gstr=0
                cv.WaitKey(1500)
                cv.DestroyWindow("Snap Shot")


def CheckSnap(xr,yr,xy,yy,xg,yg,xb,yb):
        if xb and xr and xg and xy:
```

```
        print "all colors"
        if (xr-xg)<50 and (xy-xb)<50 and (yr-yg)<50 and (yy-yb)<50:
                print "trigger snap"
                take_snap(xr,yr,xy,yy,xg,yg,xb,yb)
```

Gesture matching for Image viewer

```
def CheckImgV(xy,yy,xr,yr,xg,yg,xb,yb):
        global invoke_ShowImage
        if (yb and yg) and (not(yy or yr)):

                if yg-yb in range(50,90):
                        invoke_ShowImage+=1
                        print "show Image"
                if invoke_ShowImage==5:
                        print "invoke show Image"
                        invoke_ShowImage=-2
                        return 1

        return 0


def ShowImage(xy,yy,xr,yr,xg,yg,xb,yb):

        global frame
        global flist
        zoomed,flag=0,0
        os.system(''+flist[1]+' ~/Pictures &')
        while not flag:

                xy,yy,xr,yr,xg,yg,xb,yb=GetCoords()
                flag=CheckImgV(xy,yy,xr,yr,xg,yg,xb,yb)
                if yr and yy and not(yg or yb):
                        BackNext(yr,yy)
                if yy and yr and yg and yb:
                        zoomed=Zoom(xr,yr,xy,yy,xb,yb,xg,yg)
                if yb and yg and zoomed and not(yr or yy):
                        Move(xb,yb,xg,yg)
                ShowFrame()
```

```
        if flag:
                os.system('xdotool key "Alt+F4"')
                cv.WaitKey(200)

        cv.Flip(frame,frame,1)
        if xr and xb:
                if xr-xb in range(50,180) and abs(yr-yb)<30:
                        close+=1
                if close>=10:
                        os.system('xdotool key "Alt+F4"')
                        close=0

        if xr and xb and not(xy or xg) :
                print "exiting"
                print xr-xb
                if xr-xb in range(50,180):
                        clos+=1
                if clos>=10:
                        clos=0
                        sys.exit(0)

inc_success=0
dec_success=0

def BackNext(yr,yy):
        global inc_success,dec_success

        if yy-yr > 200 :
                print yy-yr
                inc_success+=1
                print "inc"
        elif inc_success:
                inc_success-=1
        print inc_success
        if yr-yy > 200:

                dec_success+=1
                print "dec"
                print dec_success
```

```
        elif dec_success:
                dec_success-=1

        if dec_success==10 :
                os.system('xdotool key "BackSpace"')
                cv.WaitKey(1000)
                dec_success=-2
                inc_success=-2



        if inc_success==10:
                os.system('xdotool key "space"')
                cv.WaitKey(1000)
                inc_success=-2
                dec_success=-2



zoomin,zoomout,zoomed=0,0,0
lxr,lyr,lxy,lyy,lxg,lyg,lxb,lyb=0,0,0,0,0,0,0,0

def Zoom(xr,yr,xy,yy,xb,yb,xg,yg):

        global zoomin,zoomout,lxr,lyr,lxy,lyy,lxg,lyg,lxb,lyb,zoomed

        if lxr ==0:
                lxr,lyr=xr,yr
        if lxy==0:
                lxy,lyy=xy,yy
        if lxb==0:
                lxb,lyb=xb,yb
        if lxg==0:
                lxg,lyg=xg,yg




        if lxr<xr and lxy<xy and lxg>xg and lxb>xb:
                if lyy-lyr>yy-yr and lyg-lyb>yg-yb:
                        zoomout+=1

        if lxr>xr and lxy>xy and lxg<xg and lxb<xb:
```

```
            if lyy-lyr<yy-yr and lyg-lyb<yg-yb:
                zoomin+=1

        if zoomin==2:
            os.system('xdotool key "Ctrl+plus"')
            os.system('xdotool key "Ctrl+plus"')
            zoomed+=1
            print "zoom in"
            zoomin,zoomout = 0,-2



        if zoomout==2:
            os.system('xdotool key "Ctrl+minus"')
            os.system('xdotool key "Ctrl+minus"')
            os.system('xdotool key "Ctrl+minus"')
            zoomed-=1
            print "zoom out"
            zoomout,zoomin= 0,-2



        lxr,lyr,lxy,lyy,lxg,lyg,lxb,lyb=xr,yr,xy,yy,xg,yg,xb,yb
        print zoomin,zoomout,"z"
        return zoomed

left,right,up,down=0,0,0,0 # ctrl movements
def Move(xb,yb,xg,yg):
        global left,right,up,down,lxg,lyg
        if yg-yb<100:
            return
        if lxg==0:
            lxg=xg

        if lxg>xg:
            right+=1
        elif lxg<xg:
            left+=1
        if lyg<yg:
            down+=1
```

```
elif lyg>yg:
        up+=1

if down==5:
        print "dwn"
        os.system('xdotool key "Down"')
        os.system('xdotool key "Down"')
        up,down=0,0
if up==5:
        print "up"
        os.system('xdotool key "Up"')
        os.system('xdotool key "Up"')
        up,down=0,0
if right==10:
        print "right"
        os.system('xdotool key "Right"')
        os.system('xdotool key "Right"')
        right,left=0,0
if left==10:
        os.system('xdotool key "Left"')
        os.system('xdotool key "Left"')
        right,left=0,0
        print "left"
lxg=xg
```