MeMouse

By

PC Innayatullah
PC Noor Yasin
PC Nida Waheed
NC Misbah Munir

Submitted to the Faculty of Computer Science Department
Military College of Signals, National University of Sciences and Technology,
Rawalpindi in partial fulfillment for the requirements of a
B.E Degree In Computer Software Engineering
JULY 2010

ABSTRACT

MeMouse is interactive software that has been developed to control presentations in real time environment. Various products are available to facilitate presentation that includes hardware support like digital presenters and smart boards. It has been observed in our system that the presentation screen

and computer are not placed in a way that user initiates any mouse operation instantly. It is time-consuming and it also affects the attention of the presenter as well. Hence, a concept that facilitates the presenter during presentation using his hand gestures has been presented.

Using digital image processing techniques, software has been developed that enables the user to perform the mouse operations using hand gestures. These gestures can trigger mouse operations when user is present at the point of presentation. The properties that have been used to extract hand are skin color detection are motion, using HSL values and edge detection. Tracking of hand has been made efficient using the Kalman Filter. Time dependent gestures for Single Click, Double Click and Right Click have been included. The message is passed to Windows API accordingly and operations are performed.

The software has been developed in C# .net and tested using scrum methodology. Unit, integration and system testing have been performed. Required results have been achieved more than 80% depending upon different constraints.

Acknowledgment

First of all, we thank Allah almighty for this opportunity we gained and His guidance and help.

We would like to acknowledge Col. Naveed Sarfaraz Khattak, CS Dept. MCS, for providing us with actual idea for doing this project. We would have not done anything without his guidance.

We also want to mention Dr. Imran Siddiqi for helping us in finding many solutions that were required for completing this project throughout. We also acknowledge Mr. Bashir Bilal for helping us in completion of this project. Without the support of these guides, this would have been a difficult job.

Table of Contents

List of Figures……viii

List of Tables……ix

1. Introduction to MeMouse..... 1

1.1. Introduction……...1

1.2. Background……1

1.3. Problem Addressed……2

1.4. Goals and Objectives……..2

1.5. Deliverables……3

1.6. Document Organization……3

1.7. Summary……...3

2. Related Work…… 5

2.1 Introduction……..…5

2.2 Smart Boards……....5

2.3 Interactive Whiteboards Using the Wiimote……...6

2.4 Digital Presenters……...7

2.5 Summary……..8

3. Requirement Specification……9

3.1 Introduction……9

3.2 Project Scope……9

3.3 Product Features……10

3.4 Assumptions and Dependencies……...11
3.5 System Features……12
3.6 External Interface Requirements……13
3.7 Other Non-Functional Requirements……14
3.8 Software Quality Attributes……15
3.9 Other Requirements……17
3.10 Summary……18
4. Software Design……19
4.1 Introduction……19
4.2 System Overview……19
4.3 Assumptions and Dependencies……...19
4.4 System Requirements……20
4.5 General Constraints……20
4.6 Architectural Strategies……...20
4.7 System Architecture……22
4.8 Use case……24
4.9 Class Diagram... ……26
4.10 Summary……28
5. Implementation Details…… 30
5.1 Introduction……30
5.2 Video Capture……...30
5.3 Hand Extraction……32
5.4 Tracking……36
5.5 Gesture Recognition……37
5.6 Windows API Interaction……38
5.7 Summary……38
6. Testing……40

6.1 Introduction……..40

6.2 Testing Levels……40

6.3 Box Approach……42

6.4 Summary……..…42

7. Results and Analysis……43

6.1 Introduction……..……43

6.2 Results……..……43

6.3 Analysis..……..……46

6.4 Summary……..…48

8. Conclusion and Future Work……49

7.1 Introduction……..49

7.2 Concept……49

7.3 Future Work……..49

7.4 Summary……..……50

Bibliography……..……51

List of Figures

Figure 2.1(a) Use of Smart Board in class room……6

Figure 2.1(b) Use of Smart Board in other learning environment ……6

Figure 2.2(a) Interaction using Wiimote……...7

Figure 2.2(b) Interaction using Wiimote……...7

Figure 2.3 Digital Presenter …….....8

Figure 4.1 Control flow Diagram.....23

Figure 4.2 Use Case Diagram ……25

Figure 4.3 Class Diagram of MeMouse……..27

Figure 5.1 Sobel Operators….....35

Figure 7.1 Hand Detection and tracking in MeMouse……..44

Figure 7.2 Indication for Single Click……..44

Figure 7.3 Indication for Double Click……....45

Figure 7.4 Right Click operation performed….....46

Figure 7.5 User performed Refresh operation ……...46

List of Tables

Figure 5.1 HSI values used……33

Chapter 1

Introduction to MeMouse

1.1 Introduction

MeMouse is an interactive platform that has been developed using Digital Image Processing Techniques to provide the user with mouse functionalities. Computer Vision techniques and Human Computer Interaction principles are fundamental in this scenario. This chapter describes the background, basic functionalities of the system that were required to initiate that idea and specifications according to which the system has been developed.

1.2 Background

Human Computer Interaction is an important field today. Research and Development in this field is increasing day by day. Several problems are being solved and improvements in many already provided solutions are being made. MeMouse also provides interaction with the computer. Idea of interaction has been developed by taking the idea from smart board. MeMouse focuses on letting the user interact with the presentation without

the aid of mouse and by using a camera. This software focuses on the way any system can be controlled through defined hand gestures. Hand Gestures take over the functionality of mouse in a computer while the user interacts with it, being present at the point of presentation.

1.3 Problem Addressed

While presenting, a user needs to move between the slideshow and hardware control. In our environment, computers are placed on the left side of stage and presentation is projected on the right side of the stage. If an instructor, for example, wants to explain a particular point, he moves to the projected screen. But if there is a need to change slide or perform any mouse operation then he has to come back where the computer is placed to perform any operation. This consumes time and affects the attention of the presenter as well as the audience. This inconvenience needs to be sorted. MeMouse addresses this problem in a way to provide mouse functionalities using Computer Vision Techniques.

1.4 Goals and Objective

MeMouse has been developed to facilitate the Human Computer Interaction. The objective of MeMouse is to provide hardware free mouse functionalities to the presenter using computer vision techniques. The functions that are mandatory to be performed are: single click, double click and right click. The goal of the project is to first capture the video of the presenter. From captured video, the hand of the presenter has to be

captured. Hand gestures are to be observed and matched with the defined gestures. If any gesture matching the defined gestures appears, the computer needs to perform the corresponding operations. All of these operations need to be performed in real time environment. And gestures have to be defined in a way that user is most comfortable to use.

1.5 Deliverables

The deliverable for this project is the software that will control the computer according to the gestures of the presenter. And basic slide show control options must be provided.

1.6 Document Organization

This document provides basic knowledge about MeMouse. First, description of the project has been given and next chapter explains the related work in the field and other software products providing same functionality has been discussed. This thesis also includes requirement specification of the software in chapter 3. Design specifications have also been discussed in chapter 4 of this thesis. Chapter 5 describes the implementation details of our project, after that an analysis of the software has been provided before the testing of MeMouse and a few suggestions about the enhancements of the project have also been provided.

1.7 Summary

MeMouse is an interactive platform that provides mouse functionalities under the domain of Computer Vision. This Chapter provides introduction to MeMouse, goals and objection which had been set in order to develop this software. A brief elaboration of document contents has also been added.

Chapter 2

Related Work

2.1 Introduction

This chapter provides an insight into different technologies that have already been developed for Human Computer Interaction under different domains including Signal Processing and other hardware devices. All of these devices of applications provide relevant functionality. An overview of all these technologies is presented along with a comparison with MeMouse.

2.2 Smart Boards

The Smart Board interactive whiteboard is an interactive whiteboard that uses touch detection for user input – e.g., scrolling, right mouse-click – in the same way normal PC input devices, such as a mouse or keyboard, detect input. A projector is used to display a computer's video output on the interactive whiteboard, which then acts as a large touch-screen. The components are connected wirelessly, via USB or serial cables [1]. Figure

2.1(a) and 2.1(b) shows the Smart Board being used. A projector connected to the computer displays the computer's desktop image on the interactive whiteboard. The interactive whiteboard accepts touch input from a finger, pen or other solid object. Each contact with the Smart Board interactive whiteboard is interpreted as a left-click from the mouse. Other functions are implemented in same way by using different inputs.

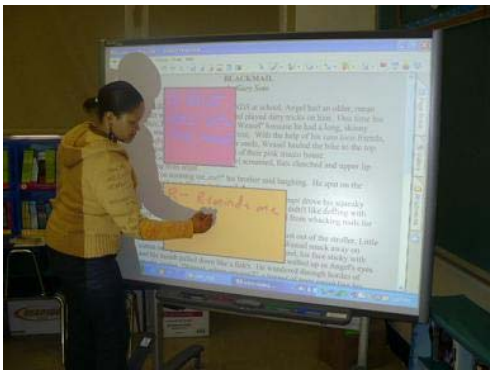

Figure 2.1(a) Use of Smart in class room

Figure 2.1(b) Use of Smart in other learning environment

MeMouse on the other hand just needs a camera that has to be connected to the computer and its position has to be in such a place where user comes in its visual field. Hence a large touch screen is not needed to be installed. This reduces the cost of system. A normal Smart Board is available in a few hundred dollars to several thousand dollars. Smart Boards are touch-sensitive devices. Accidental touch can trigger many unwanted operations on screen. MeMouse has less such drawbacks. Hence MeMouse is prioritized over Smart Boards.

2.3 Interactive Whiteboards Using the Wiimote

Wiimote is a remote controlling device that uses an infrared (IR) camera to detect the infrared light source. An IR camera is integrated in wiimote to detect energy sources. Wiimote can track sources of infrared light; user can track pens that have an IR led in the tip [2]. By pointing a wiimote at a projection screen or LCD display, user can create very low-cost interactive whiteboards or tablet displays as compared to Smart Boards. MeMouse provides even cheaper solution that uses a web camera which is far cheaper than an IR camera. Also MeMouse does not need any aiding device for detection of hand or as input. Hence it is better to use MeMouse rather than Wiimote.

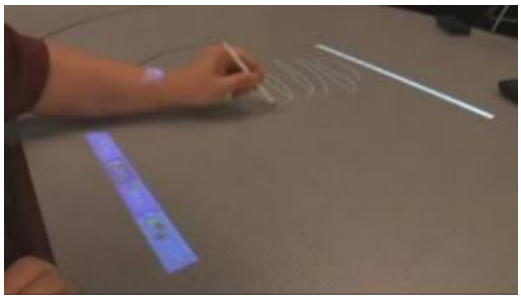

Figure 2.2(a) Interaction using a Wiimote

Figure 2.2(b) Interaction using a Wiimote

2.4 Digital Presenters

Digital presenter is a small USB like device that is connected to computer, as shown in figure 2.3. Another device that is of same size is connected to it via Bluetooth. Buttons are provided on presenter which triggers limited operations required to change slides during presentation. It can point on using a laser, in many cases along with slide show control. MeMouse on the other hand have broader scope, not only slides are controlled but

other mouse operations are performed. Hence, MeMouse gains an edge over Digital presenters.

Figure 2.3 Digital Presenter

2.5 Summary

This chapter describes the existing technologies that provide same functionalities as MeMouse. Smart Boards, Digital Presenters and Wiimote are closely similar to MeMouse, an analysis and comparison of MeMouse has been provided with all technologies. It has been observed that MeMouse is cheaper than any of these technologies and provides with same results, hence MeMouse has an edge over these devices.

Chapter 3

Requirement Specification

3.1 Introduction

This chapter describes the requirements specifications for the version 1.0 of Undergraduate degree project of software engineering. The idea of the product is a software program that helps the presenter to interact with the presentation comfortably. It captures the hand gestures of the presenter using a camera and performs the operations of the mouse. So the presenter does not have to have hardware always at his disposal. This document provides the specification that this software needs to fulfill in order to be most useful.

3.2 Project Scope

MeMouse has a limited scope that is to be able to provide the functionalities of mouse as outputs like ‘Single Click’, ‘Double Click’ and ‘Right Click’. Also basic Slide Show Control has to be provided.

All of these functionalities need to be provided using hand gestures as inputs. Hands need to be tracked and analyzed. If any hand gesture matches the defined gesture, the associated operation for that gesture has to be performed. These functions are to be provided in real time environment.

3.3 Product Features

MeMouse needs to have some specific features that ensure the usability and durability of the product. These features are the core functionalities of the system and have been useful in design of the system.

3.3.1 Recognize Hand Gestures

MeMouse can interact with Windows API as a mouse without the use of actual mouse. User can perform right click, left click and tracking on the screen. These operations can be initiated by performing some defined hand gestures that is to keep hand static for a particular time.

3.3.2 Skin Detection

MeMouse can detect the skin of any person without the aid of any color or other hardware. For this, user just needs to keep his hand in the visual field of camera.

3.3.3 Hand Recognition

MeMouse can recognize the hand in the area of visual field of camera that will find hand in the area and recognize it.

3.3.4 Hand Tracking

Hand is tracked in the image using predictions of Kalman Filter. The focus of the program is the area where hand is present in order to reduce search space.

3.3.5 Windows API interaction

MeMouse interacts with Windows API in order to pass messages about mouse operations. Messages are passed to Operating System about performing single click, double click and right click.

3.4 Assumptions and Dependencies

For the development of MeMouse, there are some assumptions have been made. This section describes the assumptions and dependencies of MeMouse based on which, the software had to be developed.

3.4.1 Basic Assumptions

It has been assume that hand is the most moving object in our environment. But there is a drawback in using this assumption that if any other object having skin color moves faster than human hand then MeMouse will classify it as Human Hand. Also that the hand does not stop movement for up to or more than 2 seconds if any operation performance is not intended.

3.4.2 Operating System

Windows family higher than windows 98 is targeted Operating System family. As it is the most used Operating Environment, it has been selected.

3.5 System Features

System features according to functional requirements provided by the Project Supervisor are as follows:

3.5.1 Capture Video

System needs to capture video so that human hand can be detected from the user environment. And certain actions can be performed based on the gesture of the hand. User has to initiate the program for video capturing. After starting the program user has to click “start” button present on the form to start capturing. And the response of the system is to display the video as output.

3.5.2 Hand Extraction

After video capturing, hand has to be extracted. So that tracking and gesture recognition can be performed. This feature has high priority because it initializes the system at first and no further processing can be done without input generated by this module. At this stage, user has to interact with camera rather than direct interaction with the program. Hence response from camera has to be accurate in order to perform the processing on video.

3.5.3 Tracking

This is an essential part of MeMouse. It has to be efficient in order to track hand on screen and cursor will move accordingly. Response of this

feature is to give expected position of the object based on present position and velocity of the object.

3.5.4 Gesture Recognition

Gesture Recognition is core requirement of MeMouse. The performed gesture has to be classified, after analysis, as legal or illegal. This has to pass message to the main class to perform the action accordingly. Response of gesture recognition has to classify the gesture and send request to main program for related action.

3.6 External Interface Requirements

MeMouse has specific interface requirements that have been discussed in this chapter.

3.6.1 User Interfaces

MeMouse is an interactive program that always needs to get input from the user. User needs to be in the defined area that is the visual field of the camera. The hand should be visible in the scenario in order to get input. The input can only be taken if the hand stays in the visual field when making a specific gesture. The gestures must be able to perform the mouse operation for the presentation. Hand must act as a mouse. The gestures defined for MeMouse to perform Single Click, Double Click, Right click and Basic Slide Show Control has been defined.

Also the hand has to be detected using the properties like skin color and hand movement, no other aiding material should be needed to do so.

3.6.2 Hardware Interfaces

MeMouse gets input through web-camera that must have high resolution in order to get clear input with less noise.

3.6.3 Software Interfaces

This product needs to interact with the operating system of the platform through API. Targeted Operating System must include Windows XP and Windows Vista.

3.7 Other Nonfunctional Requirements

Certain other functionalities are required based on performance and response of MeMouse. These requirements are described here.

3.7.1 Performance Requirements

MeMouse has to be efficient software in terms of response and operation. As the domain of the product is image manipulation that needs fast processing on the machine along with being efficient. Hence the program logic and data flow needs to be in a way to be most efficient. MeMouse needs to work under the normal lighting conditions, with non-static background, be robust, compatible with the platform and also respond within minimum time in order to produce output.

3.8 Software Quality Attributes

MeMouse has to follow some requirements that affect the quality of the system. Quality of MeMouse has to be improved by following the quality requirements described in this section.

3.8.1 Runtime System Qualities

At run time MeMouse has to adopt some function in order to provide the user with required functionalities. As system has to perform its functions in real time so, runtime qualities of MeMouse are as follows:

3.8.1.1 Functionality

MeMouse must perform the functions like right click, single click and double click at any point of the screen.

3.8.1.2 Performance

MeMouse must be able to perform operations in less time that is acceptable that is within a second.

3.8.1.3 Availability

MeMouse must be available for performance all the time user needs it to. For example in the presentation environment MeMouse must be available.

3.8.1.4 Usability

MeMouse has to be user friendly. User must be able to use MeMouse in most convenient way.

3.8.2 Non-Runtime System Qualities

Non-Runtime qualities of MeMouse are those which are required for enhancement in code or to make MeMouse useful for other developers in enhancing the system for other requirements and for other environments for which this system can be extended.

3.8.2.1 Modifiability

MeMouse has to be able to accommodate changes that include modifying MeMouse to incorporate more gestures. Also, the software must be able to accommodate any other functions if any other user, like a programmer, wants to incorporate.

3.8.2.2 Portability

MeMouse should have the ability of a system to run under different computing environments. As the targeted environment for MeMouse is presentation or lecture environment but other such environments where user want to use the system for personal use should be covered.

3.8.2.3 Reusability

MeMouse applications must to be reusable in new applications. If a system is developed which needs the functionalities of MeMouse, MeMouse should be easy to understand that could be implemented in such a way.

3.8.2.4 Integrate-ability

Separately developed components of the system have to work correctly together in MeMouse. Modules of MeMouse must collaborate with each other in such a way to perform in way to be most useful.

3.8.2.5 Testability

MeMouse must be able to be tested in order to free it from faults. Different tests including beta testing is necessary in order to remove faults and make the software perform in accordance with the requirements specified.

3.9 Other Requirements

MeMouse needs to be robust and able to manage disaster situations that arise during operation and hence work in real time efficiently. By disaster situation, it is meant that the situation in which undesired inputs are provided to the system. For example, if user's hand goes out of the frame, it should be able to manage to track it when it reappears in the frame.

3.10 Summary

This chapter describes the requirements of the system as described by Project Supervisor. It includes interface, functional and non functional requirements along with the main features required by the system. These requirements have been set after checking the feasibility of the system. These requirements have been considered as the fundamental principles for testing and standardization of the product.

Chapter 3

Requirement Specification

3.11 Introduction

This chapter describes the requirements specifications for the version 1.0 of Undergraduate degree project of software engineering. The idea of the product is a software program that helps the presenter to interact with the presentation comfortably. It captures the hand gestures of the presenter using a camera and performs the operations of the mouse. So the presenter does not have to have hardware always at his disposal. This document provides the specification that this software needs to fulfill in order to be most useful.

3.12 Project Scope

MeMouse has a limited scope that is to be able to provide the functionalities of mouse as outputs like ‘Single Click’, ‘Double Click’ and ‘Right Click’. Also basic Slide Show Control has to be provided.

All of these functionalities need to be provided using hand gestures as inputs. Hands need to be tracked and analyzed. If any hand gesture matches the defined gesture, the associated operation for that gesture has to be performed. These functions are to be provided in real time environment.

3.13 Product Features

MeMouse needs to have some specific features that ensure the usability and durability of the product. These features are the core functionalities of the system and have been useful in design of the system.

3.13.1 Recognize Hand Gestures

MeMouse can interact with Windows API as a mouse without the use of actual mouse. User can perform right click, left click and tracking on the screen. These operations can be initiated by performing some defined hand gestures that is to keep hand static for a particular time.

3.13.2 Skin Detection

MeMouse can detect the skin of any person without the aid of any color or other hardware. For this, user just needs to keep his hand in the visual field of camera.

3.13.3 Hand Recognition

MeMouse can recognize the hand in the area of visual field of camera that will find hand in the area and recognize it.

3.13.4 Hand Tracking

Hand is tracked in the image using predictions of Kalman Filter. The focus of the program is the area where hand is present in order to reduce search space.

3.13.5 Windows API interaction

MeMouse interacts with Windows API in order to pass messages about mouse operations. Messages are passed to Operating System about performing single click, double click and right click.

3.14 Assumptions and Dependencies

For the development of MeMouse, there are some assumptions have been made. This section describes the assumptions and dependencies of MeMouse based on which, the software had to be developed.

3.14.1 Basic Assumptions

It has been assume that hand is the most moving object in our environment. But there is a drawback in using this assumption that if any other object having skin color moves faster than human hand then MeMouse will classify it as Human Hand. Also that the hand does not stop

movement for up to or more than 2 seconds if any operation performance is not intended.

3.14.2 Operating System

Windows family higher than windows 98 is targeted Operating System family. As it is the most used Operating Environment, it has been selected.

3.15 System Features

System features according to functional requirements provided by the Project Supervisor are as follows:

3.15.1 Capture Video

System needs to capture video so that human hand can be detected from the user environment. And certain actions can be performed based on the gesture of the hand. User has to initiate the program for video capturing. After starting the program user has to click “start” button present on the form to start capturing. And the response of the system is to display the video as output.

3.15.2 Hand Extraction

After video capturing, hand has to be extracted. So that tracking and gesture recognition can be performed. This feature has high priority because it initializes the system at first and no further processing can be done without input generated by this module. At this stage, user has to

interact with camera rather than direct interaction with the program. Hence response from camera has to be accurate in order to perform the processing on video.

3.15.3 Tracking

This is an essential part of MeMouse. It has to be efficient in order to track hand on screen and cursor will move accordingly. Response of this feature is to give expected position of the object based on present position and velocity of the object.

3.15.4 Gesture Recognition

Gesture Recognition is core requirement of MeMouse. The performed gesture has to be classified, after analysis, as legal or illegal. This has to pass message to the main class to perform the action accordingly. Response of gesture recognition has to classify the gesture and send request to main program for related action.

3.16 External Interface Requirements

MeMouse has specific interface requirements that have been discussed in this chapter.

3.16.1 User Interfaces

MeMouse is an interactive program that always needs to get input from the user. User needs to be in the defined area that is the visual field of the camera. The hand should be visible in the scenario in order to get input.

The input can only be taken if the hand stays in the visual field when making a specific gesture. The gestures must be able to perform the mouse operation for the presentation. Hand must act as a mouse. The gestures defined for MeMouse to perform Single Click, Double Click, Right click and Basic Slide Show Control has been defined.

Also the hand has to be detected using the properties like skin color and hand movement, no other aiding material should be needed to do so.

3.16.2 Hardware Interfaces

MeMouse gets input through web-camera that must have high resolution in order to get clear input with less noise.

3.16.3 Software Interfaces

This product needs to interact with the operating system of the platform through API. Targeted Operating System must include Windows XP and Windows Vista.

3.17 Other Nonfunctional Requirements

Certain other functionalities are required based on performance and response of MeMouse. These requirements are described here.

3.17.1 Performance Requirements

MeMouse has to be efficient software in terms of response and operation. As the domain of the product is image manipulation that needs fast

processing on the machine along with being efficient. Hence the program logic and data flow needs to be in a way to be most efficient. MeMouse needs to work under the normal lighting conditions, with non-static background, be robust, compatible with the platform and also respond within minimum time in order to produce output.

3.18 Software Quality Attributes

MeMouse has to follow some requirements that affect the quality of the system. Quality of MeMouse has to be improved by following the quality requirements described in this section.

3.18.1 Runtime System Qualities

At run time MeMouse has to adopt some function in order to provide the user with required functionalities. As system has to perform its functions in real time so, runtime qualities of MeMouse are as follows:

3.18.1.1 Functionality

MeMouse must perform the functions like right click, single click and double click at any point of the screen.

3.18.1.2 Performance

MeMouse must be able to perform operations in less time that is acceptable that is within a second.

3.18.1.3 Availability

MeMouse must be available for performance all the time user needs it to. For example in the presentation environment MeMouse must be available.

3.18.1.4 Usability

MeMouse has to be user friendly. User must be able to use MeMouse in most convenient way.

3.18.2 Non-Runtime System Qualities

Non-Runtime qualities of MeMouse are those which are required for enhancement in code or to make MeMouse useful for other developers in enhancing the system for other requirements and for other environments for which this system can be extended.

3.18.2.1 Modifiability

MeMouse has to be able to accommodate changes that include modifying MeMouse to incorporate more gestures. Also, the software must be able to accommodate any other functions if any other user, like a programmer, wants to incorporate.

3.18.2.2 Portability

MeMouse should have the ability of a system to run under different computing environments. As the targeted environment for MeMouse is

presentation or lecture environment but other such environments where user want to use the system for personal use should be covered.

3.18.2.3 Reusability

MeMouse applications must to be reusable in new applications. If a system is developed which needs the functionalities of MeMouse, MeMouse should be easy to understand that could be implemented in such a way.

3.18.2.4 Integrate-ability

Separately developed components of the system have to work correctly together in MeMouse. Modules of MeMouse must collaborate with each other in such a way to perform in way to be most useful.

3.18.2.5 Testability

MeMouse must be able to be tested in order to free it from faults. Different tests including beta testing is necessary in order to remove faults and make the software perform in accordance with the requirements specified.

3.19 Other Requirements

MeMouse needs to be robust and able to manage disaster situations that arise during operation and hence work in real time efficiently. By disaster

situation, it is meant that the situation in which undesired inputs are provided to the system. For example, if user's hand goes out of the frame, it should be able to manage to track it when it reappears in the frame.

3.20 Summary

This chapter describes the requirements of the system as described by Project Supervisor. It includes interface, functional and non functional requirements along with the main features required by the system. These requirements have been set after checking the feasibility of the system. These requirements have been considered as the fundamental principles for testing and standardization of the product.

Chapter 4

Software Design

4.1 Introduction

This chapter provides with the design specifications of MeMouse. These specifications have been developed using the requirements described in previous chapter. This chapter provides information regarding system structure and architecture.

4.2 System Overview

MeMouse is software that is intended to facilitate the user while they are delivering presentation. In order to control the presentation there are

several way that includes manual interaction through mouse or keyboard, use of digital presenter or smart board. This creates a lot of inconvenience during presentation and distracts the audience which in turns wastes time. But while using MeMouse the presenter will be able to interact with the presentation at the point where output is being projected.

4.3 Assumptions and Dependencies

Basic assumption that has been the basis of development is that presenter's hand is the most moving object in MeMouse's environment. Another assumption has been that Windows Operating System released after Windows 98 are used in presentation environment.

4.4 System Requirements

For better performance the system on which MeMouse runs should fulfill requirements like having Pentium 4 or above (C2D recommended), 512 MB RAM (Recommended) and Graphics card (optional)

4.5 General constraints

MeMouse is efficient software that provides output in different scenarios but there are some conditions that have to be applied in order to get usability from the system. Following are the constraints that have been applied

1 The room must have sufficient light so that skin color can be recognized.

2 Hand should move more than any other body part initially, particularly in 2 seconds at the start of program.

3 Full sleeve shirts are recommended for best performance. Cloth color must not have color that matches skin.

4 There must not be any other skin colored object in the background.

4.6 Architectural Strategies

Design decisions and strategies that affect the overall organization of the system have been described here, higher-level structures of system. Some important issues are describes in this section like language, platform and project extensions.

4.6.1 C# Platform:

C# has been used to develop MeMouse. Main reason to use C# is that the application is being developed in real-time and requires fast execution. Another reason is that it is widely being used for Image processing techniques. It is better than MATLAB because MATLAB is efficient for manipulation of still images and does not produce efficient results when used in real-time and videos.

4.6.2 Library

An open source library AForge.net is being used for image processing techniques. It is the best open source library for image processing with

C#. Another choice is Open CV that is not being used because of having some compatibility issues when using with C#.

4.6.3 Future plans

At present, main focus is to control slideshows using hand gestures. But there is a plan to extend the functionality to take over mouse functionalities of computer interaction using gestures. Also, to build an application integrated with MeMouse which will act as interactive whiteboard.

4.7 System Architecture

To perform mouse operations using MeMouse, hand tracking and gesture recognition is necessary. In order to perform hand gesture recognition there are certain steps which have been followed. For that, user needs to be present in the visual field of camera. His hand has to be extracted and tracked. Gestures that hand shows are to be analyzed. Keeping in view these steps MeMouse was divided into five modules Video capture, Hand extraction, Tracking, Gesture recognition and Control Windows message passing.

The flow of data throughout the system has been shown diagrammatically in figure 4.1. When MeMouse starts its execution, it captures the video of presenter first. All processing depends upon the video that has been captured in real time. The input has to be hand gesture therefore hand is extracted from the video which was converted into frames before this

operation performance. Next step is to track hand, in order to move cursor and finding next input. If hand shows defined gesture, a message to Windows API is passed to perform a particular operation. All of these modules have been explained in detail in this section afterwards.

4.7.1 Hand extraction

Hand extraction is not a standalone and single task rather it is a series of different tasks which are Skin detection, Edge detection, Motion residue and Blob finding. After getting the resulted images from skin detection, edge detection and motion residue a logical AND is applied on these images. The resulted blob is considered as the hand.

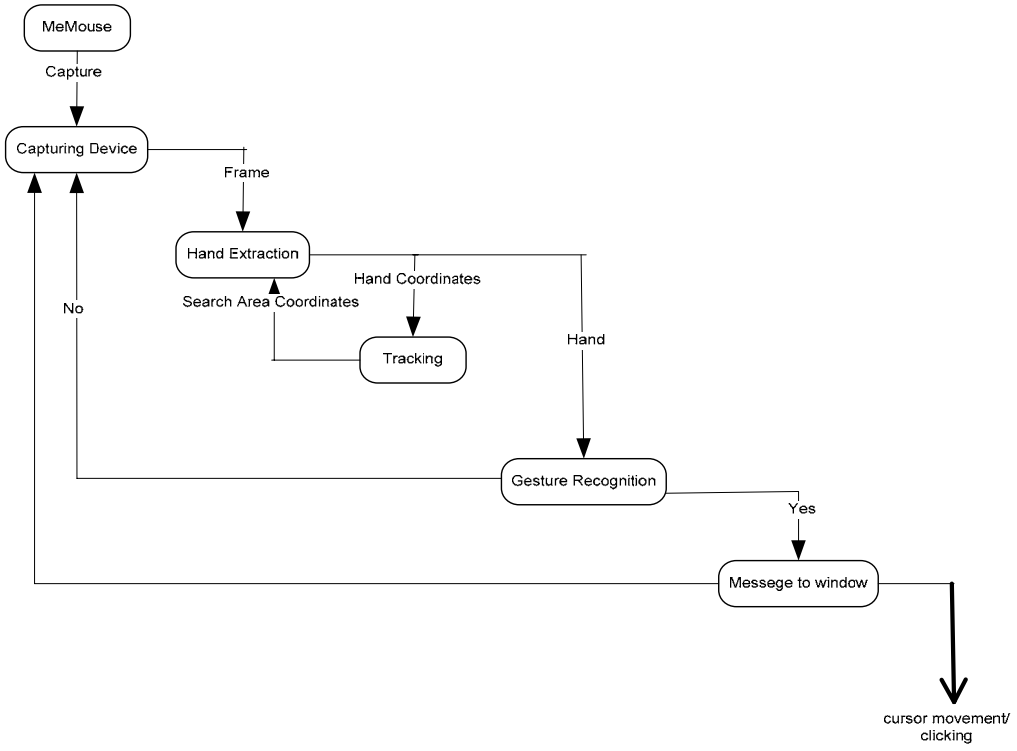

Figure 4.1 Control Flow Diagram

4.7.2 Tracking

After identifying the hand, it is being tracked. Kalman filter is used to predict the next position of the hand based on the present position. When next position is predicted skin colored objects are identified in the squeezed window (that is the hand).

4.7.3 Gesture Recognition

Gestures that had to be defined were supposed to be most convenient to use. Hence, time dependent gestures have been defined. System clock puts a check about the time hand is kept static on a particular area of screen. Notifications appear about clicking option and operation is performed after hand moves after being kept static for a particular time.

4.8 Use case

The use case diagram of the system has been given as Figure 4.2. This diagram describes the interaction of user and the system.

4.8.1 Basic Flow of System

User places his hand in front of camera; camera takes the image and sends it to clip board where it is saved. System captures new frame and sends a copy of frame to detect skin; skin module sends the extracted image back to the clip board. System sends another copy to detect edges; module returns extracted edges in frame. System sends current and previous frame to motion residue module to calculate motion in

subsequent frames; module returns a frame with difference of both the frames. System performs Logical AND of all three returned frames; then Blog counting is done to extract the maximum blog from the resulted frame which is ‘Hand’ of the user. Systems sends center of hand to filter to predict possible position of hand in next frame and also sends hand shape for gesture recognition. A filter takes position and predicts next position. Gesture recognition module processes the shape to find out the gesture; if gesture found system sends message to windows API to perform action, else next frame is captured. Next frame is searched in a restricted area obtained from Kalman Filter output.

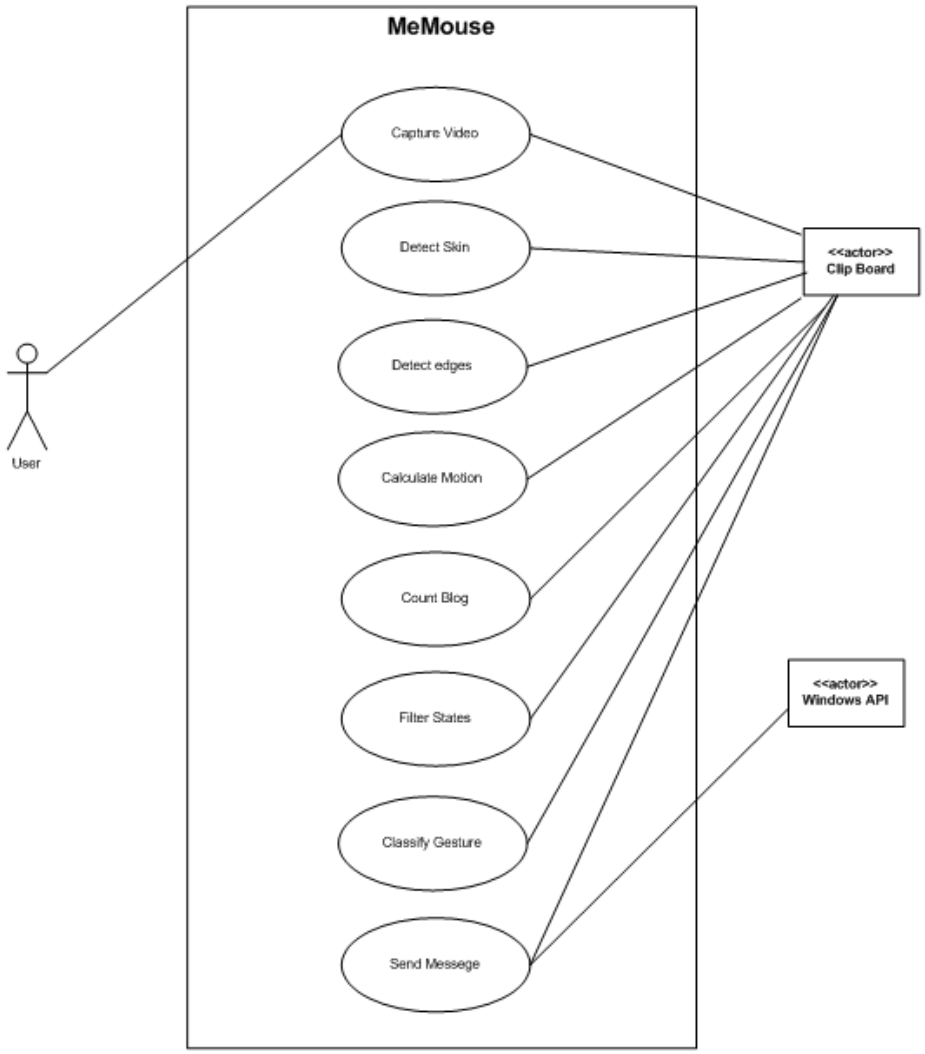

Figure 4.2 Use Case Diagram

4.8.2 Post Conditions:

System has performed the action which was requested by user through hand gesture.

4.8.3 Alternate Scenarios:

There is no alternate scenario because the objective is defined and product is being developed by strictly following the requirements.

4.9 Class Diagram

Class Diagram of MeMouse has been presented as Figure 4.3 and elaborated in this section. All the classes in the diagram are described briefly. And a legend is provided in the diagram to describe the symbols that have been used and their purpose.

4.9.1 MeMouse

As it has been shown in Figure 4.3, MeMouse is the main class. This is the class which controls all other classes and interacts with them in order to perform required functionality. When user first starts the program, user's direct interaction with the program is over now it is the responsibility of MeMouse class to carry out further actions and procedures.

4.9.2 WebcamCapture

MeMouse class shifts the control to WebcamCapture class to capture video and recognize gesture. WebcamCapture class contains the objects of KalmanProcessing KalmanProperties and MotionDetector3. These objects are used to interact with the respective classes.

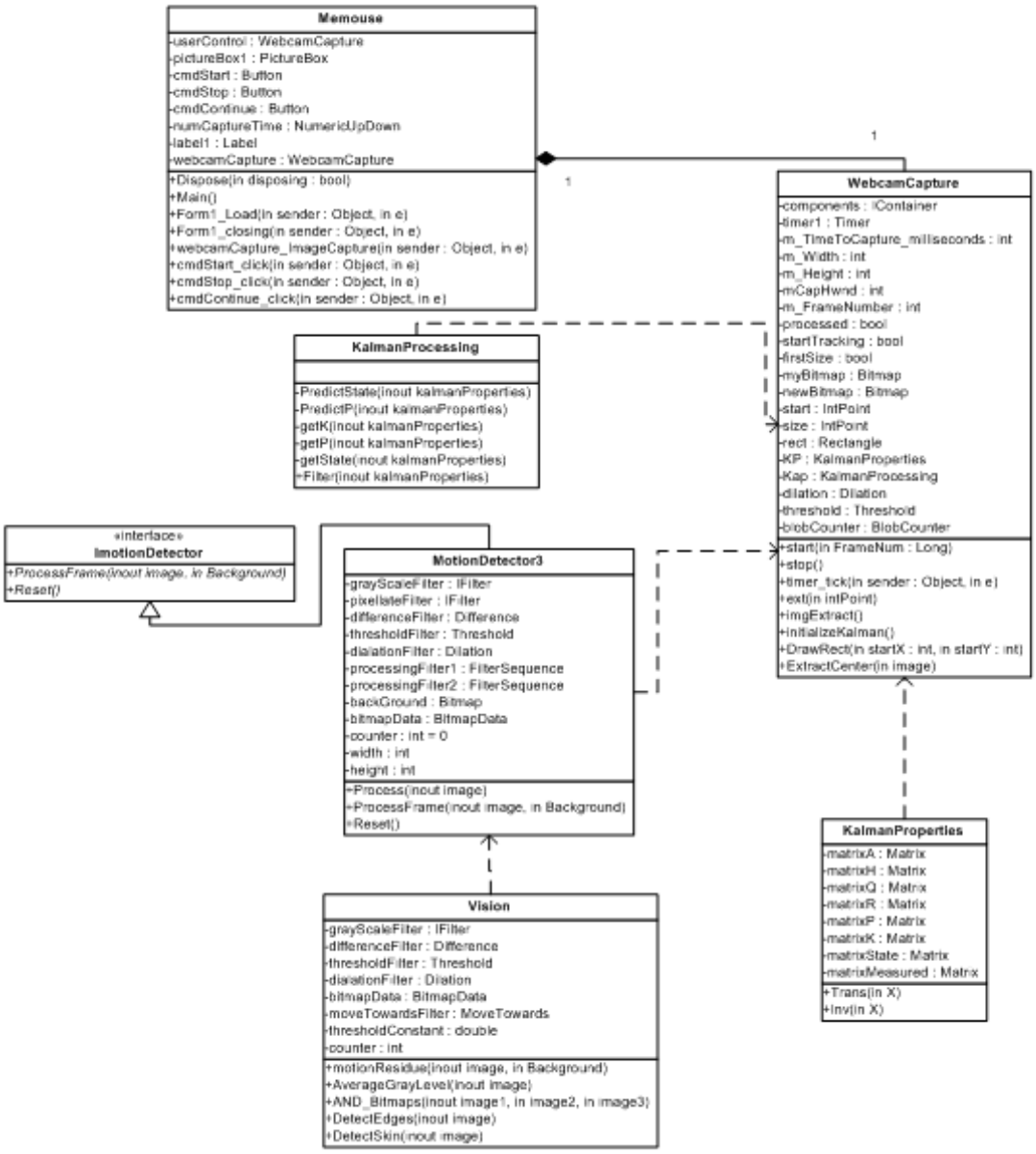

Figure 4.3 Class Diagram of MeMouse

4.9.3 KalmanProperties

KalmanProperties class contains the data and functions which are required for predicting the track for an object in the scene. It simply defines data variables and makes it available for other classes i.e.

KalmanProcessing to be used for prediction of moving object in the frames.

4.9.4 KalmanProcessing

This class is responsible for predicting the track of object and object itself in the scene. It contains methods for prediction of next position of the object.

4.9.5 MotionDetector3

It is the implementation of interface IMotionDetector. This class contains methods for processing frames and extraction of hand in a frame. It also has the object of vision class which is used to call the functions like skindetection, edgeDetection and motion residue and others from vision class.

4.9.6 Vision

Vision class has necessary methods for finding the required attributes in any image, from skin detection to motion detection and logical AND of bitmap images.

4.10 Summary:

MeMouse has to perform operations in real time environment that is why it has to be properly designed to improve efficiency. This chapter elaborated the design of the software in accordance with the assumption and

constraints that have been applied for development. Class Diagram, Data Flow Diagram and Use Case have been added and explained in order to have better understanding of system functionalities.

Chapter 5

Implementation Details

5.1 Introduction

This chapter provides with the summary of different approaches used by people to address the problem statement of MeMouse. All of these approaches are useful but differ in efficiency and response. Complete system of MeMouse has been subdivided into five modules or subsystems based on previous technologies and team effort. These modules includes: Video Capturing, Hand Extraction, Hand Tracking, Gesture Recognition and Windows API Interaction. Hand Extraction part can be further subdivided into Skin Color Detection, Edge Detection and Residue Image. Gesture Recognition can be further subdivided into Time-based Gestures and Up/Down Gestures.

5.2 Video Capture

The first and the most important step to start with MeMouse is to capture video for real time processing. As the software will have to perform operations according to hand gestures rather than mouse itself. These

hand gestures are to be captured or have to be seen. So video of the presenter is to be captured and converted into a format that can be operated on by other system modules. As we know that a camera captures video at different rates depending upon quality and resolution of camera. We are using a web camera because it is cheap in cost and maintenance. The video that is captured can be converted into images. As C# platform does not give any functionality to convert the video in usable data, there is a need to use some other tool or technique that do so. There are two tools we considered that provide these functionalities that are AForge.NET and OpenCV. Both of these tools are available as open source software.

OpenCV has mainly been written in C and provides Digital signal processing portability. Its wrappers for C++, C# and Java are available. As C# is our development language, so it is the main focus to see the compatibility with C#. Program needs to convert the video captured in .avi format into images. In this scenario, OpenCV does not provide some important functions that were required. Hence there is incompatibility of OpenCV with C# programming interface.

AForge.NET is also an open source library for image forging. Unlike OpenCV, AForge.NET is a C# framework designed for developers in the field of computer vision and artificial intelligence [3] [4]. Hence, provides complete functionality for image manipulation in C# programming environment. AForge.NET was chosen for “Video Capture” module of

MeMouse. AForge.NET had to be used for real time image manipulation. The image processing functions by AForge.NET can be utilized by using the library AForge.Imaging. AForge.Imaging is a library for image processing routines and files. It can be used to convert the video in the image frame according to the frame capture rate. As MeMouse has to process images in real time using C#, Aforge.NET provides the best solution for the image capturing and manipulation.

5.3 Hand Extraction

Before the actual processing starts, human hand needs to be detected. MeMouse depends upon the movement of hands and hand gestures. Hand Extraction techniques help in extraction of human hand from the image and subtraction of background image. This approach has been developed using image segmentation techniques that include Skin Color Detection, Edge Detection and Motion Detection techniques. These techniques, when put together, give us Human Hand in the video. Hence, skin detection techniques could be used in order to detect hands. But as our system needs real time results so, the use of techniques provided by Khurshid and Vincent and Askar et al. [5] [6] have been used in collaboration that use hand segmentation techniques in real time.

5.3.1 Skin Detection

Skin Color is the most important property through which we can detect hand. Skin has a specific range of colors varying from region to region as

well as lightning conditions. There are also various models that can be used to find skin color that include RGB, normalized RBG, TSL, YCrCb and HSI Color Spaces. All of these approaches are efficient in different scenarios. The most important two approaches are RGB and HSI. RGB space can be used in this scenario but the main disadvantage of using this approach is that it detects a range of many other colors that are not required but the system. If we implement it with improvements that do not give errors, efficiency of the system decreases. HSI does not show inefficiency in his case. It is proven that irrespective of different races human color falls into the finite subset of HUE value. Keeping this in view and based on experimental results, HSI color space for skin detection has been chosen. Literature shows the range of HSI values for human body falls into finite subset of real values. This information was the basis of implementation of skin color detection sub module. Skin Color values of hue and saturation are more important factors to be considered. Using HSI model, better results have been achieved.

Table 5.1 HSI values used

Property Name	**Values Range**
Hue	4 to 35
Intensity (luminous)	0 to 0.7
Saturation	0 to 0.8

Using only Skin Color detection technique, hand cannot be detected efficiently. Major constraint that implies is that face is also detected in this

scenario. So other techniques are also needed to be used in collaboration with Skin Detection. There is also a constraint that applies in using this approach is omnipresence of skin colored objects in the background because those objects will be considered as skin as well. Resulted image is converted into binary image for further manipulation.

5.3.2 Edge Detection

Along with Skin Detection, another technique relevant in this case is Edge Detection. Separation of hand from other objects is made easy by finding edges. It is an important technique for finding any object in the given space also called as image segmentation. Edges of the object vary depending upon the shape of the object. Edge Detection is used for feature detection and feature extraction, which aim at identifying points in a digital image at which the image brightness changes sharply or more formally has discontinuities. There are various techniques that can be used for Edge Detection in images or videos. These techniques include Sobel Operator, Differential Edge Detection, Canny Edge Detector, Prewitt operator and methods by Roberts cross. Applying Threshold is another technique that can be applied in a way to find edges in the system. Based on these techniques, different methods have been used to find edges in the videos. Most accurate results have been obtained by using Sobel Operator. Resulted image is converted into binary image for further manipulation. Sobel Operators [7] that have been used are shown is Figure 5.1.

-1	0	+1
-2	0	+2
-1	0	+1

Gx

+1	+2	+1
0	0	0
-1	-2	-1

Gy

Figure 5.1 Sobel Opertors

5.3.3 Residue Image

Presenter uses his hands the most during presentation. Hence hands become the most moving object in the video. So in order to detect hands another property of hand movement is useful. For this purpose residue of image is found that detects motion in sequence of frames. Difference between two frames can be taken in order to detect motion. Two images are considered as matrices and mobile objects are found by examining the gray level changes in the video sequence. Let Fi(x,y) be the ith frame of the sequence, then the residue image Di(x,y) is a binary image formed by the difference of ith and (i+1)th frame to which a threshold is applied. This has been done in order to extract motion from complex backgrounds.

5.3.4 AND Image

Results of ‘Skin Detection’, ‘Edge Detection’ and ‘Residue Image’ collaboratively gives the Hand extracted from the video. As all three images are binary, common result areas in three images is hand extracted in the video sequence. All images collaboratively provide a ‘Combined’ image [5]. We find the largest contour area and its center and then draw a bounding box of fixed width and length which represents the hand region we were looking for. Further operations are to be performed on the area bound in this box.

5.4 Tracking

Tracking of hand is an important part of MeMouse. Tracking algorithm shows the cursor its position on screen. Heuristic search in complete frame for finding hand makes the software inefficient. Hence, some technique is needed to be used that makes tracking efficient. Kalman Filter [8] is used to predict the next position of moving object using basic equation of motion. This reduces the search space in frames. Kalman Filter can be implemented using second equation of motion that is:

$$x_k = Ax_{k-1} + Bu_{k-1} + w_{k-1}$$

The Kalman filter estimates a process by using a form of feedback control: the filter estimates the process state at some time and then obtains feedback in the form of (noisy) measurements. As such, the equations for the Kalman filter fall into two groups: time update equations and measurement update equations. The time update equations are responsible for projecting forward (in time) the current state and error covariance estimates to obtain the a priori estimates for the next time step. The measurement update equations are responsible for the feedback, that is, for incorporating a new measurement into the a priori estimate to obtain an improved a posteriori estimate.

Hence, time update equations are predictor equations, while the measurement update equations are corrector equations. Indeed the final estimation algorithm resembles that of a predictor-corrector algorithm for solving numerical problems. Time Update equations of Kalman Filter are:

$$x1_k = Ax1_{k-1} – Bu_{k-1}$$

$$P1_k = AP_{k-1}A^T + Q$$

Measurement Update equations of Discreet Kalman Filter are:

$$K_k = P_k .H^T(HP_k . H^T + R)$$

$$x2_k = x1_k + K_k (z_k – Hx1_k)$$

$$P2_k = (I - K_kH). P1_k$$

After each time and measurement update pair, the process is repeated with the previous a posteriori estimates used to project or predict the new a priori estimates. The Kalman filter recursively conditions the current estimate on all of the past measurements.

5.5 Gesture Recognition

User needs to show a hand gesture that triggers the mouse operation on the computer. There can be several types of gestures that can be used. Time dependent gestures are the simple in implementation as well as convenient for the user. Hence a timer has been integrated in order to recognize user's input. Time constraint has been added depending upon the usage of the input. User needs to keep his hand static in MeMouse's scenario in order to initiate mouse operation. As single click is most frequently used, hence only 2 second time is required to trigger this

operation. After that, double click is the most frequent mouse operation, for which 4 seconds timer has been set. And 6 seconds for right click. User is informed if he keeps his hand static for a particular time, and options are notified on screen. When user moves his hand after keeping it static, operation is performed according to input.

5.6 Windows API Interaction

Finally there is a need to interact with the windows API that will trigger operations on the computer. This can be done using the functions defined in Windows API [9]. The functions we need the most for the use of mouse include tracking of mouse cursor, managing single and double clicks. A class for Windows interaction has been added to perform the core functionality for the system.

5.7 Summary

Implementation details of MeMouse have been discussed in this chapter. Techniques like skin color detection, edge detection, motion detection, hand extraction and Kalman filter have been discussed. Process starts with video capture, extracts hand using motion, skin color and edge

detection. After that tracking is performed in order to move the cursor and perform functions to capture inputs. Depending upon time, gestures are recognized and operations are performed accordingly.

Chapter 6

Testing

6.1 Introduction

To ensure the use and quality of the product, testing is conducted. Accuracy of output produced by MeMouse has to be tested and maintained in order to improve the quality of the software. Software Testing techniques used and therefore obtained results of success are discussed in this section.

6.2 Testing Levels

Different modules were developed separately in order to provide different functionalities of MeMouse. All of the modules were tested at different levels and also after integration. The levels at which MeMouse has been tested are discussed in this section.

6.2.1 Unit Testing

Each Module was developed and tested at unit level. The results were obtained by using different data sets. In the case of MeMouse, the data sets were the use of software using hand gestures. Therefore testing was done accordingly. At unit level, testing was done for every module. The module for video capturing was developed first, that was tested by capturing video at five different systems. And 100% results were obtained. The module of Hand Extraction was tested by 15 different users out of which 13 hands were detected. Overall this module was showing 86.7% accurate results. Kalman filter was applied and was tested on hypothetical values. The results obtained were 93% accurate. Hence the implementation was carried forward. The Gesture Recognition Module was implemented and tested. The results obtained at first were not up to the expectations. It was further implemented and then out of 20 test cases 17 provided accurate results. Hence 85% accuracy was obtained. And Windows API interaction module was tested and it provided with 100% accurate results. The slide show control module was tested by 13 people and gave more than 90% results.

6.2.2 Integration Testing

Integration testing was performed in order to completely ensure the stability of the system and performance in order to be maximum useful. Also the errors introduced due to integration of two modules are removed using this technique. The modules were integrated in order of their implementation. The integration testing gave more than 90% accurate results. Hence, the errors introduced in MeMouse due to integration of different modules were minimized.

6.2.3 System Testing

System testing for MeMouse was performed at the end of development and integration. Complete system was tested in three different places having almost same conditions for light and background. The final system testing results were accurate in accordance with their inputs. More than 80% test cases were successful. That is a major achievement that will help in further enhancement of system.

6.3 Box Approach

Box Approach testing is the testing that deals with the functionality of the system in accordance with the written code. There were two types of Box

Testing Techniques we used for MeMouse. The details of both techniques are given in this section.

6.3.1 Black Box Testing

Black Box Testing was carried out at every level of testing by inserting dummy outputs at different points. And the errors introduced were removed.

6.3.2 White Box Testing

White Box testing was carried out at unit, integration and system levels. All errors noticed were removed. White Box testing took most of the time in testing overall.

6.4 Summary

Testing is quite important in maintaining quality and enhances usability of the product. Different types of testing techniques were used at different levels to ensure that product works accurately and efficiently. All errors that were detected were removed.

Chapter 7

Results and Analysis

7.1 Introduction

MeMouse has been developed to work in real time environment. This is a way to control presentation in real time using computer vision techniques rather than single processing and/or other similar ways that includes touch sensitive or using devices that operate using wireless signals or Bluetooth.

7.2 Results

MeMouse has been developed in a way to facilitate users while delivering a presentation. The idea has been to facilitate the user feel comfortable by not making him use any aiding device or material. The milestone has been achieved to control slideshow during presentation. A series of snapshots presented in this chapter gives better understanding of the results we achieved.

Hand Detection is an important milestone, for which image processing techniques have been used. These techniques help in the detection of hand to perform further operations. Figure 6.1 shows the hand detection during the operation of MeMouse. A rectangular window covers the area where hand is detected by MeMouse. This has been achieved by using

Skin detection, EDGE Detection and Blob Finding and then getting a combined image that gave the similarity that is hand. As the Figure 6.1 shows, hand has been detected which is the major milestone to find input in the scenario.

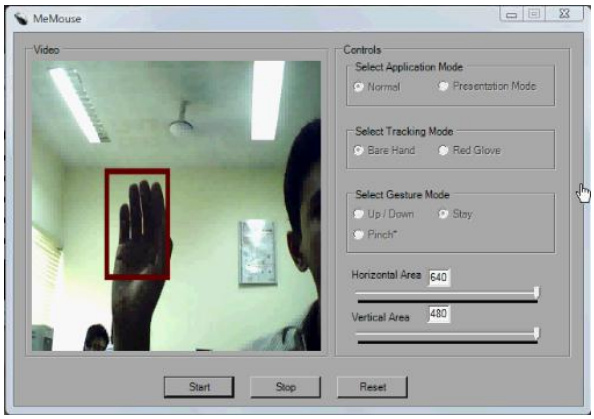

Figure 6.1 Hand Detection and tracking in MeMouse

Tracking of hand is another important issue that has been achieved in efficient way. Using Kalman Filter tracking has been made better. Onscreen tracking is an important step as it spots the point where user needs to perform operation. Tracking of hand shows the output on screen simultaneously. As the hand moves, cursor moves on computer screen.

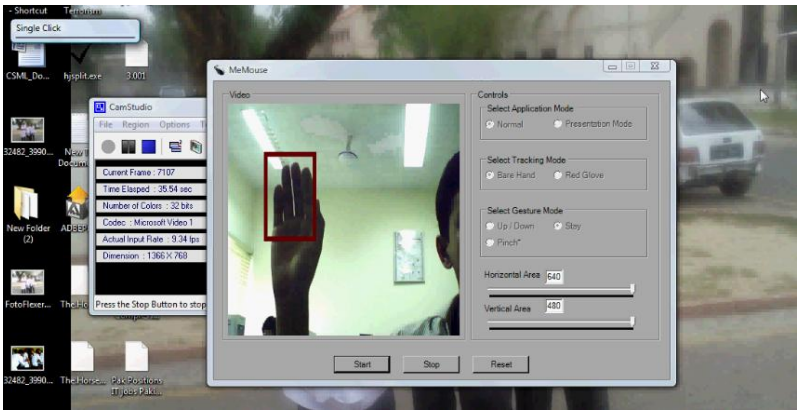

Figure 6.2 Indication for Single Click

Static Gestures have been added in the system to perform any operation. If the user keeps his hand static for 2 seconds, user is notified that if he moves his hand now, Single Click operation will be performed. Figure 6.2 shows the indication of Single Click to the user. User has his hand static for 2 seconds now. But if user wants to perform Double Click Operation, he has to keep his hand static for 2 more seconds. Overall 4 seconds is the time for which user needs to keep his hand static in order to perform Double Click operation. Figure 6.3 show that user is notified about that if he moves his hand now Double Click operation will be performed.

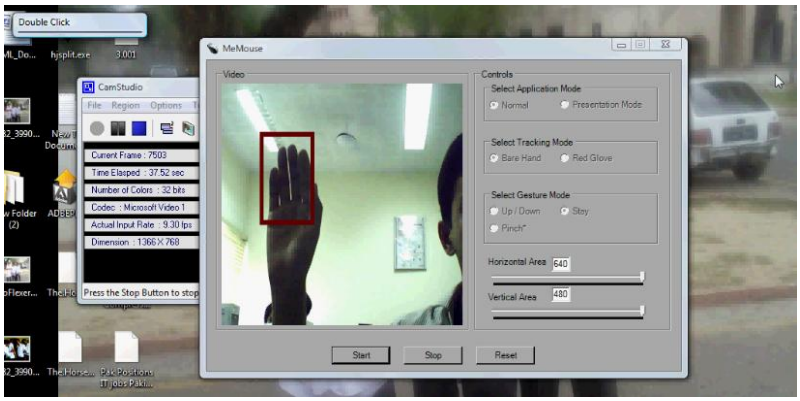

Figure 6.3 Indication for Double Click

If the user needs to perform Right Click on screen, 6 seconds is the minimum time the user has to keep his hand static. After that an indication appears on screen that if user moves his hand, Right Click operation will be performed. Figure 6.4 depicts the operation performed if user keeps his hand static for 6 seconds. Like previous two cases, a notification precedes the performance of Right Click operation. User usually needs to perform Single Click after performing Right Click. Figure 6.5 show that user has performed ‘Refresh’ operation by Right Clicking on desktop and then Single Clicking on Refresh tab.

Figure 6.4 Right Click operation performed

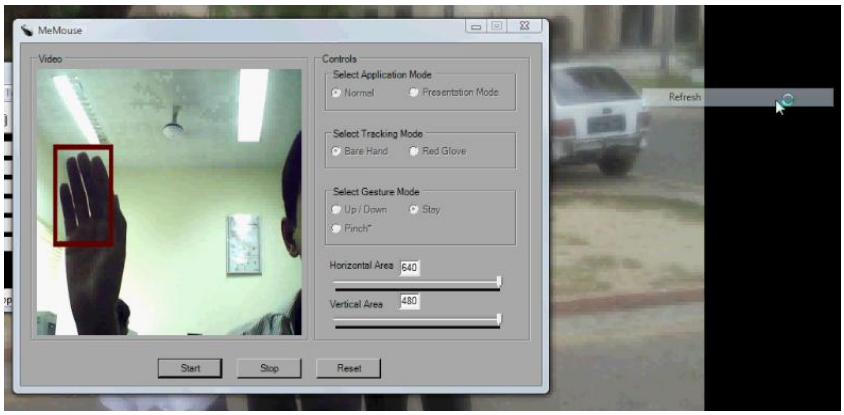

Figure 6.5: User performed Refresh operation

7.3 Analysis

Other techniques that are used to control slide show includes Smart Boards, Wiimote and Digital Presenters. These are the devices that need more hardware support than MeMouse.

7.3.1 Digital Presenter

Digital Presenter is a device, connected via Bluetooth Technology to its other half in order to provide limited functionalities of Mouse. User needs to attach a Bluetooth connecting device with the computer in order to perform operations. The hardware required is expensive as well as it costs more if any fault happens. Digital Presenter controls the slides during presentation but cannot perform if any other operation is to be performed like opening any other program. User is restricted to a particular program. Whereas MeMouse can provide more functions that is Mouse functionality that can be used to open and close other programs as well during presentation.

7.3.2 Smart Board

Smart Board is a useful device. It has similar functionality as of MeMouse, but it has to have a large touch screen in order to perform Mouse operations. The equipment of Smart Board is quite expensive that is difficult to purchase for many institutes and organizations. Hence, MeMouse provides a cheaper solution to the same problem. MeMouse

provides the functionality of Mouse operations in real time which makes MeMouse a better option than a Smart Board.

7.3.3 Wiimote

Wiimote is a device that is uses IR camera in order to perform the functions. Also it needs the user to use a pen-like device that has Light Emitting Diode (LED) at its tip. When that LED emits light, Wiimote tracks it and perform operations. On the other hand, MeMouse does not need any aiding material and provides the same functionalities in terms of Mouse operations. Also, Wiimote is expensive device than a webcamera that we use for MeMouse in order to get input. Hence, MeMouse is a smart choice.

7.4 Summary

MeMouse can perform the operations like Singe Click, Double Click and Right Click which in turns helps to control the Slide Show. Some of these devices provide even less functions than MeMouse being expensive as compared to MeMouse. Functionality provided by MeMouse is same as other technologies similar to it in much cheaper and hence efficient way.

Chapter 8

Conclusion and Future Work

8.1 Introduction

This chapter describes the overall achievements of MeMouse. Also, some suggestions have been presented n order to enhance the system and for up gradation of MeMouse. MeMouse can be extended smartly in order to cover a broader domain of Human Computer Interaction.

8.2 Concept

Concept of MeMouse was developed by the concept of Smart Boards and Digital Presenters. These devices are expensive as well as needs more expenses for their maintenance. Also, these devices are hardware dependent whereas MeMouse provides the use with freedom and not to be dependent upon hardware. MeMouse provides more convenient and useful environment in order to perform same functions which are required while presentations.

8.3 Future Work

MeMouse is just an initiation of a wide field of Human Computer Interaction during presentation. It can achieve a lot of milestones which can make a presenter feel more comfortable with the environment while presenting.

MeMouse can be extended to provide on-screen keyboard to the users. That can be done by defining coordinates according to keyboard in video capturing area. This can let the user even process documents on-screen.

There is another suggestion to extend the software in order to detect human hand motion and captures it to write on projected screen. This idea is similar to writing on white board. Techniques to track the input object along with algorithms to enhance ability to learn of software can be used. Artificial Intelligence techniques can be most useful.

MeMouse can also take over computer control in a way for which user can use it by interacting with it as Smart Board. Digital image processing techniques can be used along with Artificial Intelligence to develop a complete computer system.

All of these techniques have to work with Windows API to perform desired implementations.

8.4 Summary

Human Computer Interaction can be made more and more convenient. Computer Vision and Digital Image Processing has to work in collaboration with Artificial Intelligence and miracles can be made reality. There is a wide range of option available in order to make Human Computer Interaction more and more comfortable.