Bansilal Ramnath Agarwal Charitable Trust's

Vishwakarma Institute of Information Technology

Pune – 411048.

Project Report

On

ProGestureCursor

Submitted By,

Anand Kulkarni	B80394272
Himanshu Tyagi	B80394247
Khursheed Ali	B80394268
Snehal Govekar	B80394240

Department of Computer Engineering,
Bachelor of Engineering (Computer Engineering)
University of Pune.

Year 2012-2013

CERTIFICATE

This is to certify that the following students of B.E. Computer, Vishwakarma Institute of Information Technology, Pune.

Anand Kulkarni (B80394272)
Himanshu Tyagi (B80394247)
Khursheed Ali (B80394268)
Snehal Govekar (B80394240)

have successfully completed the project on

ProGestureCursor

in the partial fulfillment of the requirements for the completion of B.E. in Computer Engineering in 2012-2013 as prescribed by the University of Pune.

Internal Guide	**External Guide**	**H.O.D.**
(Prof. Y. K. Sharma)		(Prof. S. R. Sakhare)

Principal
(Dr. B. S. Karkare)

Department of Computer Engineering
BRACT's Vishwkarma Institute of Information Technology, Pune – 48.

ACKNOWLEDGMENT

Apart from us, the success of this report depends largely on the encouragement and guidelines of many others. We are especially grateful to our guide, Prof. Y. K. Sharma, who has provided guidance, expertise and encouragement.

We express our heartfelt gratefulness to Prof.(Mr.) S.R. Sakhare (Head of Computer Department, VIIT) for his stimulating supervision whenever required during our project work. We are very much thankful to him for providing us resources whenever we needed them without any delay.

We also express our sincere gratitude and whole hearted thanks to our Project Coordinator Prof. (Mrs.) L.A. Deshpande, for her co-operation and overall support. Her deadlines were a driving force for this project.

We are also thankful to the staff of Computer department for their cooperation and support. For helping us even after the college hours and that too without any complaints. He can never forget such a gesture.

We would like to put forward our heartfelt acknowledgement to all our classmates, friends and all those who have directly or indirectly provided their overwhelming support. They belief and confidence in us, gave us the motivation to take this project at higher levels.

- Anand, Himanshu, Khursheed, Snehal

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Contents:

Chapter **1**

Introduction

1.1 OVERVIEW

With the need of employing natural communication between humans and machines there has been a great emphasis on Human-Computer-Interaction (HCI) lately. Hand being the most effective general-purpose tool because of its dexterity, adopting hand gesture as an interface in HCI has become very common. . Currently, electro-mechanical or magnetic sensing devices such as data gloves[2] are the most effective tools for capturing hand gestures. It will not only allow the deployment of a wide range of applications in sophisticated computing environments such as virtual reality systems and interactive gaming platforms, but also benefit our daily tasks by controlling interfaces via gestures.

1.2 BRIEF DESCRIPTION

This project uses vision based hand gesture recognition system[3], to provide HCI through projected computer screen. Today when we need to use projector, its tedious to interact with computer system at the same time. One needs a help of another person to control the computer while he/she uses the projector on the dais. Solutions to specific problems (Presentations, Gaming etc.) are available in this context. But the present problem specific solutions have their limitations and are not cost effective as well. Our project proposes solution to control the entire computer using hand gestures.

1.3 PROBLEM DEFINITION

Controlling projected computer system using hand gesture recognition technique.

1.4 APPLYING SOFTWARE ENGINEERING APPROACH

The incremental build model is a method of software development where the model is designed, implemented and tested incrementally (a little more is added each time) until the product is finished. It involves both development and maintenance. The product is defined as finished when it satisfies all of its requirements. This model combines the elements of the waterfall model with the iterative philosophy of prototyping.

Figure 1.1: Incremental build model

The product is decomposed into a number of components, each of which is designed and built separately (termed as builds). Each component is delivered to

the client when it is complete. This allows partial utilization of the product and avoids a long development time. It also avoids a large initial capital outlay and subsequent long waiting period. This model of development also helps ease the traumatic effect of introducing a completely new system all at once.

The series of releases is referred to as increments, with each increment providing more functionality to the customers. After the first increment, a core product is delivered, which can already be used by the customer. Based on customer feedback, a plan is developed for the next increments, and modifications are made accordingly. This process continues, with increments being delivered until the complete product is delivered.

Chapter 2

Literature Survey

Chapter **2**

Literature Survey

The existing technologies similar to our project are stated below :

1. **Color band based hand tracking system :**

   One of the existing technologies present is Color band Hand Tracking System (CbHTS)[4]. It is slightly different than the traditional mouse. User's fingers will have different colored bands on them. CbHTS scans specific zone for color bands, and the dimension of scan zone is determined by applying fuzzy logic on cursor speed of previous image. One of the user's fingertips is treated as the mouse cursor, and single click is defined as the contact of two fingers. Double click is defined as continuous two single clicks. CbHTS concludes the centre of each finger band; it proceeds to the Judgment of clicks by analyzing the difference between time periods of the Consecutive clicks.

Figure 2.1: Color band based hand tracking system

But in this method, colors of bands and background play a vital role. Thus, it requires a camera and an algorithm which can differentiate between color bands and background color precisely. It requires that the surrounding colors

should not be same as that of the color bands and the band colors should have noticeable difference in their RGB levels. Also only predefined colors can be used in this type of tracking system.

2. **Computer control using LED stick :**

   This is another technology that is used for gesture recognition in front of the projector. A stick used in this method has a LED attached to its tip. When this stick is moved in front of the projected screen, the camera detects glowing LED and moves the cursor to the respective location on the screen. The stick has two buttons, one for left-click and another for right-click. When these buttons are pressed, the respective actions are performed at the location pointed by the stick.

   This technique is expensive to implement, since it requires number of hardware components. Also it is not flexible to use a stick every time. Color of the LED matters in this vision based tracking system as in some cases its color may not get detected because of the background's color.

3. **Kinect based hand tracking :**

   Kinect[5] has a 3-D camera, with high depth sensing capability; it is the source to various exciting systems in human-machine interaction. One potential application of Kinect technology is in the 'playing games with hand motion' environment where a person can play games on computer without physically touching consoles etc. With Kinect, processing is fast and it gives the higher efficiency. But affording it for the general purpose is very expensive.

   (a) Microphone array - Four mikes pinpoint where voices or sounds are coming from while filtering out background noise.

   (b) IR emitter - Projects a pattern of infrared light into a room. As the

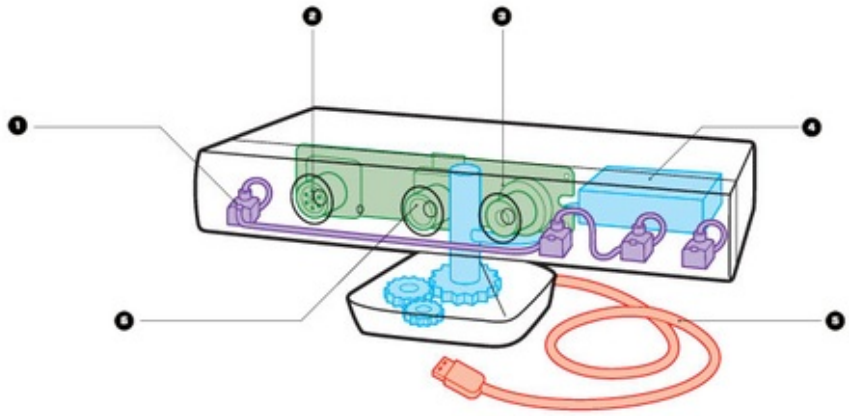


Figure 2.2: Kinect technology[1]

light hits a surface, pattern becomes distorted, and the distortion is read by the depth camera.

(c) Depth camera - Analyzes IR patterns to build a 3-D map of the room and all objects and people within it.

(d) Tilt monitor - Automatically adjusts based on the object in front of it. If you're tall, it tilts the box up. If you're short, it knows to angle down.

(e) USB cable - Transmits data to the Xbox via an unencrypted feed, which makes it relatively easy to use the kinect with other devices.

(f) Color camera - Like a webcam, this captures a video image. The kinect uses that information to get details about objects and people in the room.

Chapter 3

Software Requirement Specification

Contents:

Chapter **3**

Software Requirement Specification

3.1 INTRODUCTION

3.1.1 Purpose

The purpose of this project is to put forward a user friendly system which will reduce the dependence on input devices like keyboard or mouse while giving presentations.

3.1.2 Intended audience and reading suggestions

The users of our software will be teachers giving presentations to the students. External faculties can use this software to conduct seminars as well. This project is not restricted to faculty members only. Businessmen can also use this software for business presentations. Sales analysts can effectively explain their annual statistical reports using this software.

The introduction section explains the purpose and project requirements; both functional and nonfunctional.

The Analysis model describes flow of our project and also the implementation plan.

Next we discuss about System design using various UML diagrams. UML diagrams help to picture the system and its working clearly in terms of implementing this project. Fifth section is about the technologies we used to implement our project.

3.1.3 Project scope

The traditional method of controlling projected computer system by using keyboard or mouse clicks consumes time as the presenter has to walk all the way to the computer and then do the changes. Another option for this would be, that an assistant would be appointed to carry the whole presentation process. So we propose system that will eliminate the need of using computer peripherals while presenting seminars/lectures. The software will require a standard web camera which will be mounted near the projector. The web camera will capture images

of the projected screen at regular intervals. These images will be processed and the results[6] [7] [8] will be used to recognize hand gestures[9] made by the user. One can change a slide, click or double click on the icons, folders or can scroll the window by merely standing beside the projected screen and making corresponding gestures. This project will aid in presenting lectures, conducting seminars, business presentations, statistical reports etc.

3.1.4 Design and implementation constraints

- Hardware Constraints -
  - Projector Screen: The projector screen should have plain (color) background preferably a white colored background.
  - Web Camera :
    - Resolution : Web Camera with at least 8MP Resolution.
    - Position : The web camera should be placed in such a way that it can capture the projected image.
  - Projector Screen height: Height of the projection should be sufficient for the user , so that he/she can comfortably reach to each of the corners on it.
- Software Constraints -
  - Settings related to camera and screen should be updated whenever the camera or projector is re-positioned.
- Other Constraints -
  - The user should use only one hand while making hand gestures.
  - Only one user should use the software at a time.

3.1.5 Assumptions and dependencies

- Assumptions
  - No sudden changes in surrounding light.
  - Room is adequately illuminated.
- Dependencies
  - Brightness of projected screen.
  - Resolution of Web Camera.
  - Distance between projector and projected screen.

3.2 SYSTEM FEATURES

3.2.1 Application features

- **Intuitive -** HomePage of our software provides user with a facility of configuring the Settings .It also lets the user to Preview the previously performed settings.

Figure 3.1: HomePage to Configure Settings

- **Appealing -** ProGestureCursor software is designed in such a way that it enhances user experience with translucent windows, help option on each window and customized buttons.

- **Help Guidelines -** This software being user friendly provides help guidelines to user on each window.

- **Standardized -** Like most of the softwares in the industry, user can access number of options even from the taskbar icon. The software when minimized, minimizes to the taskbar. One can right-click on the taskbar to access various options.

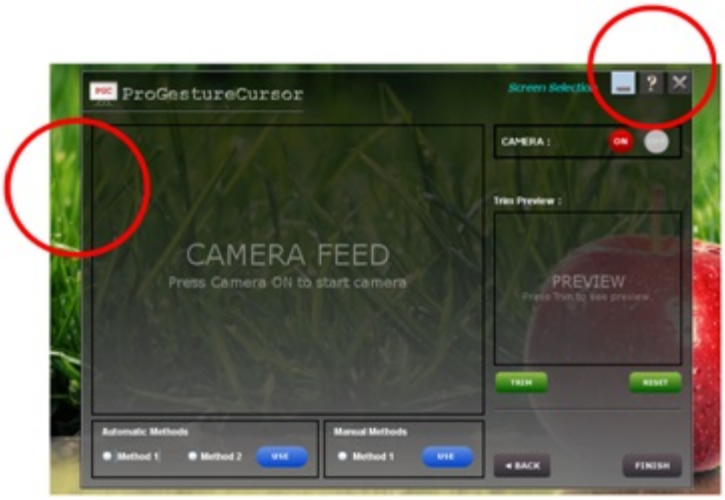


Figure 3.2: Translucency of GUI

Figure 3.3: Help messages being displayed when clicked on Help icon

Figure 3.4: Software window minimizes in Taskbar

- **User Driven -** User is provided with the option of selecting gestures of his own choice through an interactive GUI. User can modify mostly all the features of the system through GUI only. These features include various cursor options, selecting shortcuts of user's choice, choosing whether or not the alert messages should be shown and many more.

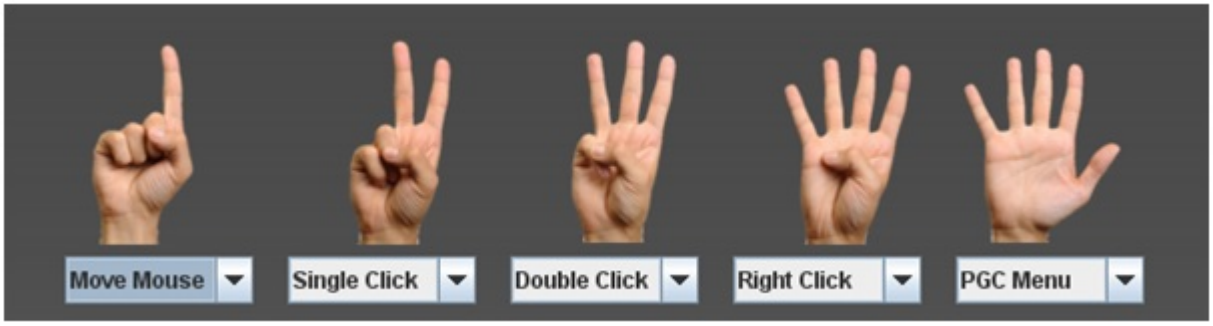


Figure 3.5: Gesture Settings

3.3 EXTERNAL INTERFACE REQUIREMENTS

- User Interfaces
  - Frontend software : Designed Application software.
  - Backend software : Text Files - storing settings.
- Hardware Interfaces
  - RAM : 1 GB (minimum)

  - WebCamera : 8MP (minimum)
  - Projector Type : Long Throw

- Software Interfaces
  - Operating System : WindowsXP/7/8, Linux
  - Additional Softwares : OpenCV(for JavaCV), Java(Jdk1.7)

3.4 NON-FUNCTIONAL REQUIREMENTS

- Performance Requirements
  - Static requirements:
    * Number of terminals-1
    * Number of simultaneous users-Only 1 user can carry the presentation using hand gestures.
    * Type of information to be handled- Images Captured via Webcam and printscreen().
  - Dynamic Requirements:
    * Images shall be processed in a good frame rate(approx. 30fps).
- Safety Requirements And Security Requirements
  - If position of camera is changed, user will have to perform the system setup again.
  - User should use only one hand to perform gestures while using the software.
  - In case of an in-efficient light conditions, hand wont be recognized by the web camera causing software to run into an idle mode.

3.5 SOFTWARE QUALITY ATTRIBUTES

- **Conceptual Integrity:**

  Conceptual integrity defines the consistency and coherence of the overall design. This includes the way that components or modules are designed, as well as factors such as coding style and variable naming.

  We have used Hungarian Notation for while coding all modules.

  Hungarian Notation is perhaps the most well-known notation, which encodes either the purpose or the type of a variable in its name. Methods are written in lowerCamelCase pattern that is, with the first letter lowercase and the first letters of subsequent words in uppercase.

- **Maintainability:**

  Maintainability is the ability of the system to undergo changes with a degree of ease.

  User can change shortcuts settings, preview settings without having to make changes in the code.

- **Reusability:**

  Reusability defines the capability for components and subsystems to be suitable for use in other applications and in other scenarios.

  Modules that are required repetitively are written in such way that they can be accessed easily. It helped in reducing Lines of Code (LOC) which effectively reduced implementation time.

- **Performance:**

  Performance is an indication of the responsiveness of a system to execute any action within a given time interval.

  Latency: - Software checks for minimum 5 frames before invoking an event corresponding to the gesture made by the user.

- **Reliability:**

  Reliability is the ability of a system to remain operational over time.

  Software starts processing captured images only when it detects a motion in-front of the web camera, Otherwise the software remains in idle mode.

- **Supportability:**

  Supportability is the ability of the system to provide information helpful for identifying and resolving issues when it fails to work correctly.

  Our software shows important messages at the right-bottom corner of the screen.

  E.g. If our software failed to retrieve previous settings It shows No Settings Found. Message at the right-bottom corner of the screen.

3.6 SYSTEM IMPLEMENTATION PLAN

The system is implemented in four main modules as follows:

1. Screen detection
2. Hand detection
3. Gesture detection
4. Call to Windows APIs

3.7 ANALYSIS MODEL

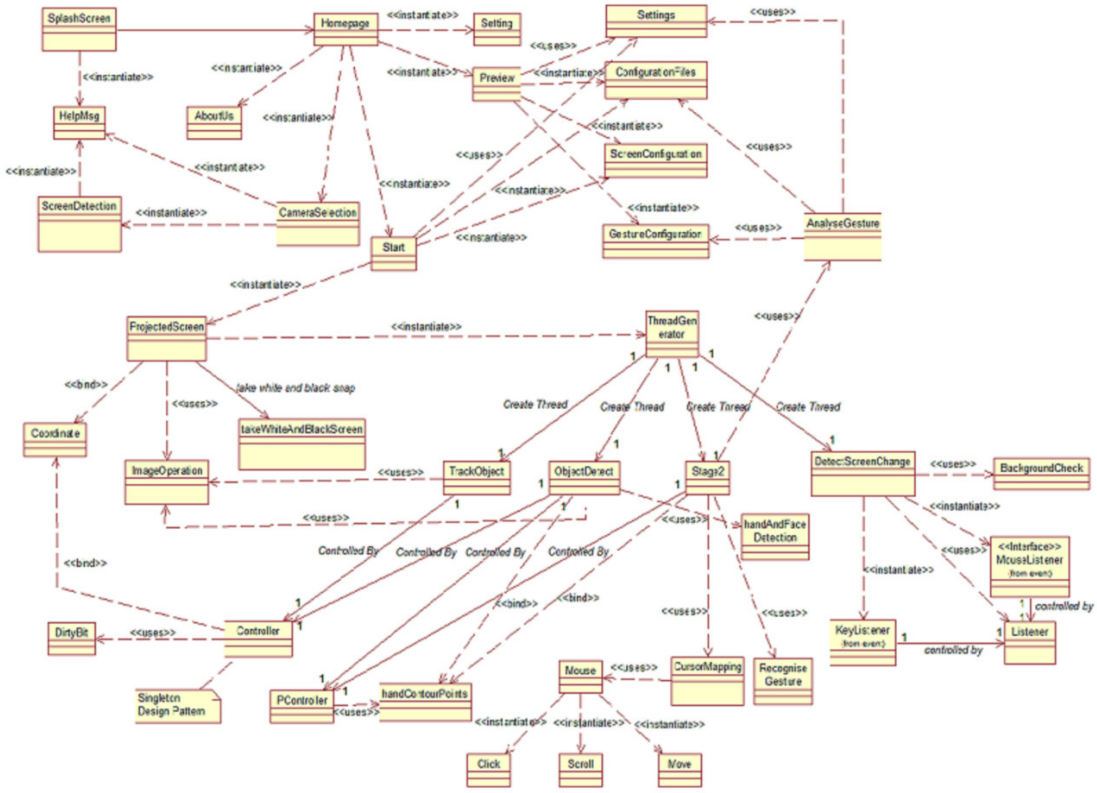


Figure 3.6: Class Diagram

Chapter 4

System Design

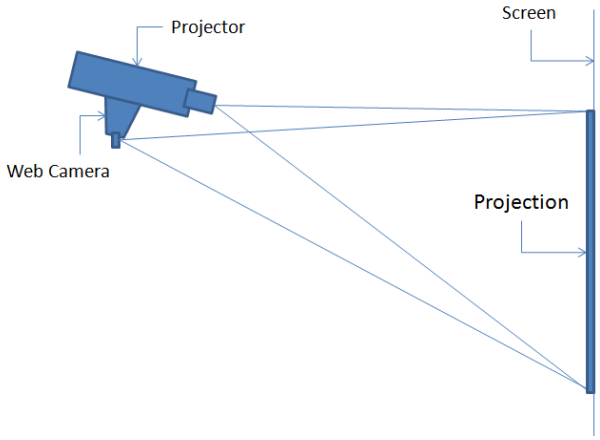
Contents:

Chapter **4**

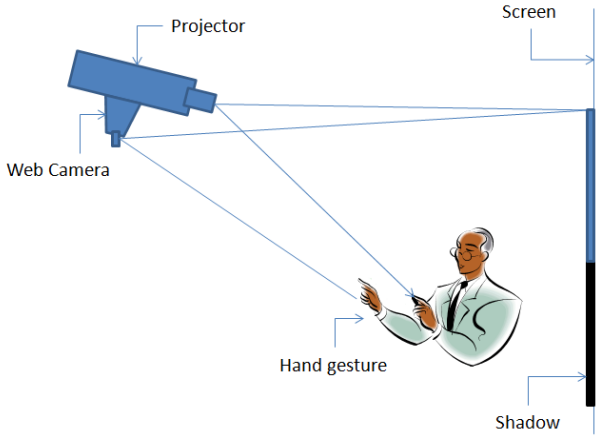
System Design

4.1 SYSTEM ARCHITECTURE

Figure 4.1 (a) and (b) shows the hardware setup and how user interacts with it. A projector is placed at a sufficient distance from the projected screen. Web camera is placed near or on top of projector from where it can capture an entire image of projected screen properly. The distance between web camera and projector screen is flexible. As shown in Figure 4.1 (b). User performs hand gestures in-front of the projected screen and these images are used for further processing.



(a) System



(b) System with User

Figure 4.1: System Architecture And usage

4.2 UML DIAGRAMS

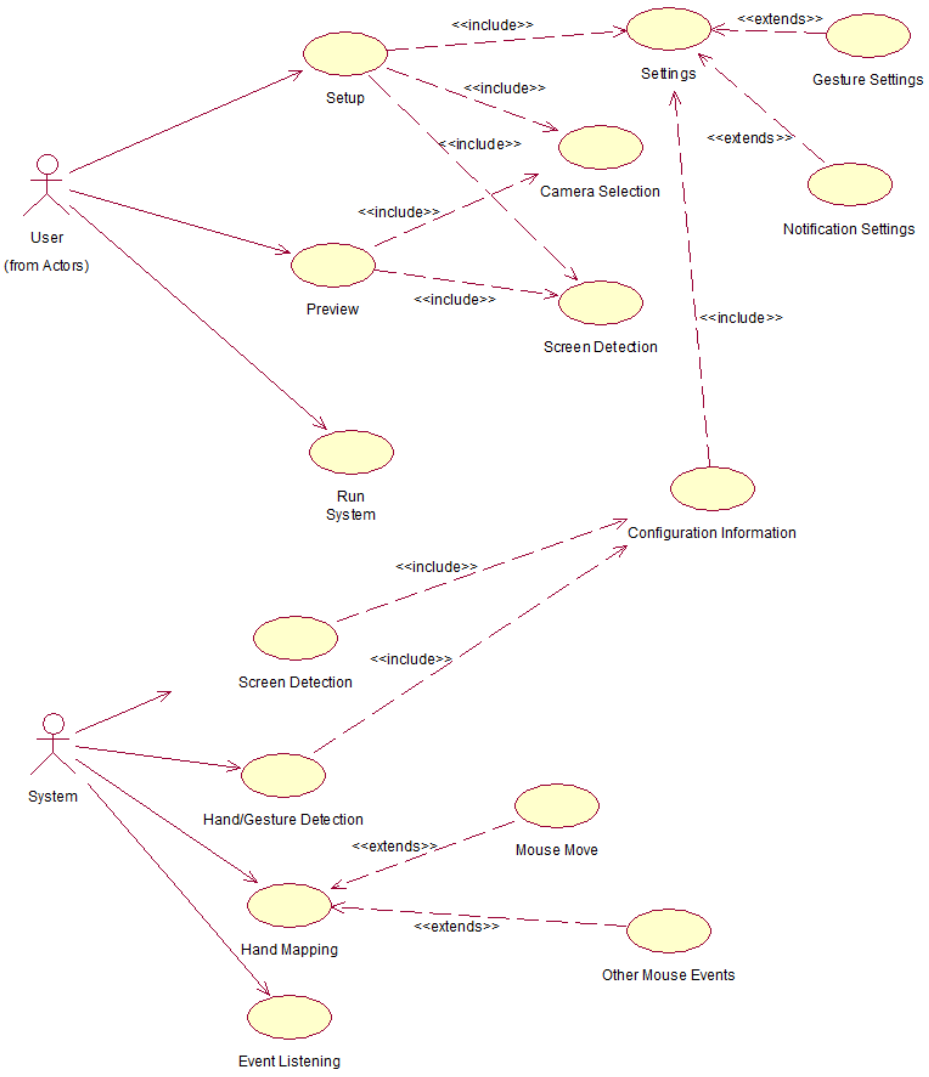


Figure 4.2: Use Case Diagram

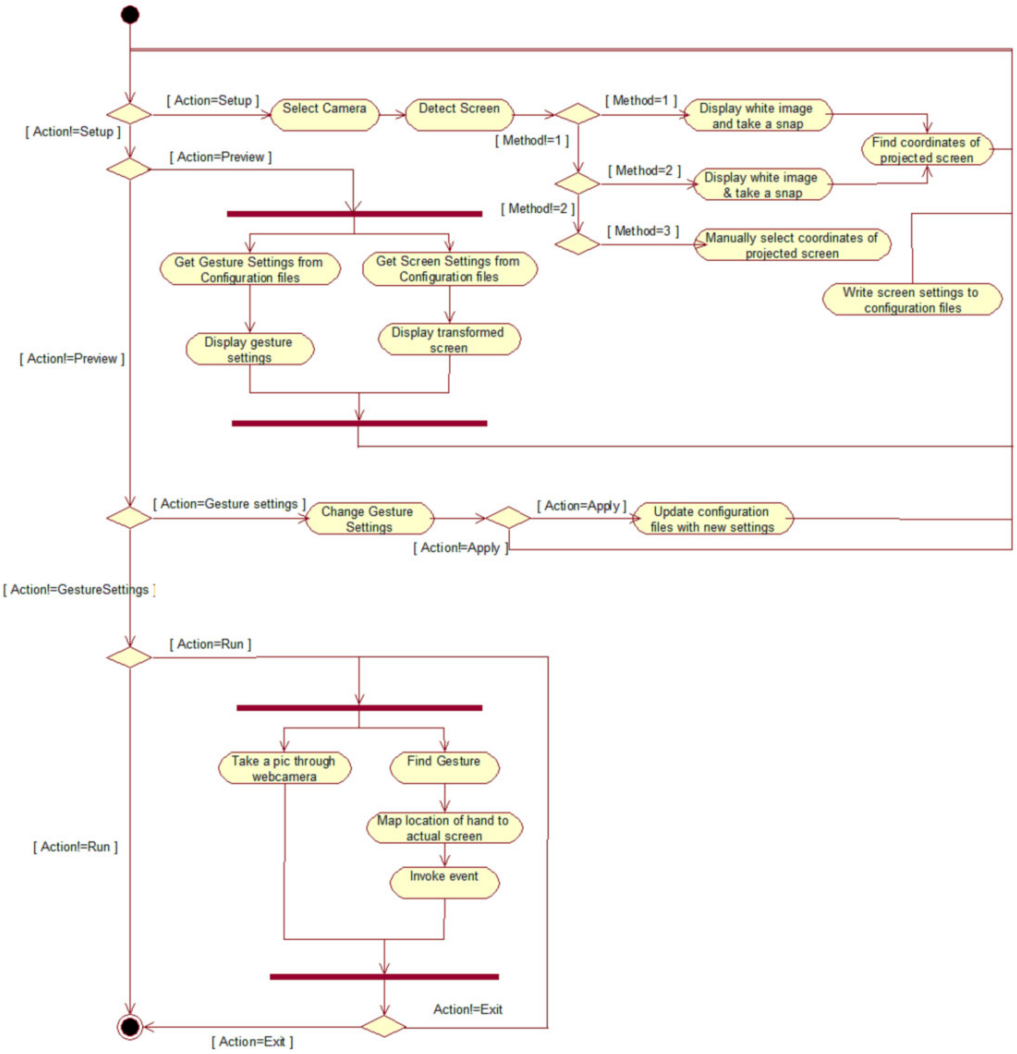


Figure 4.3: Activity Diagram

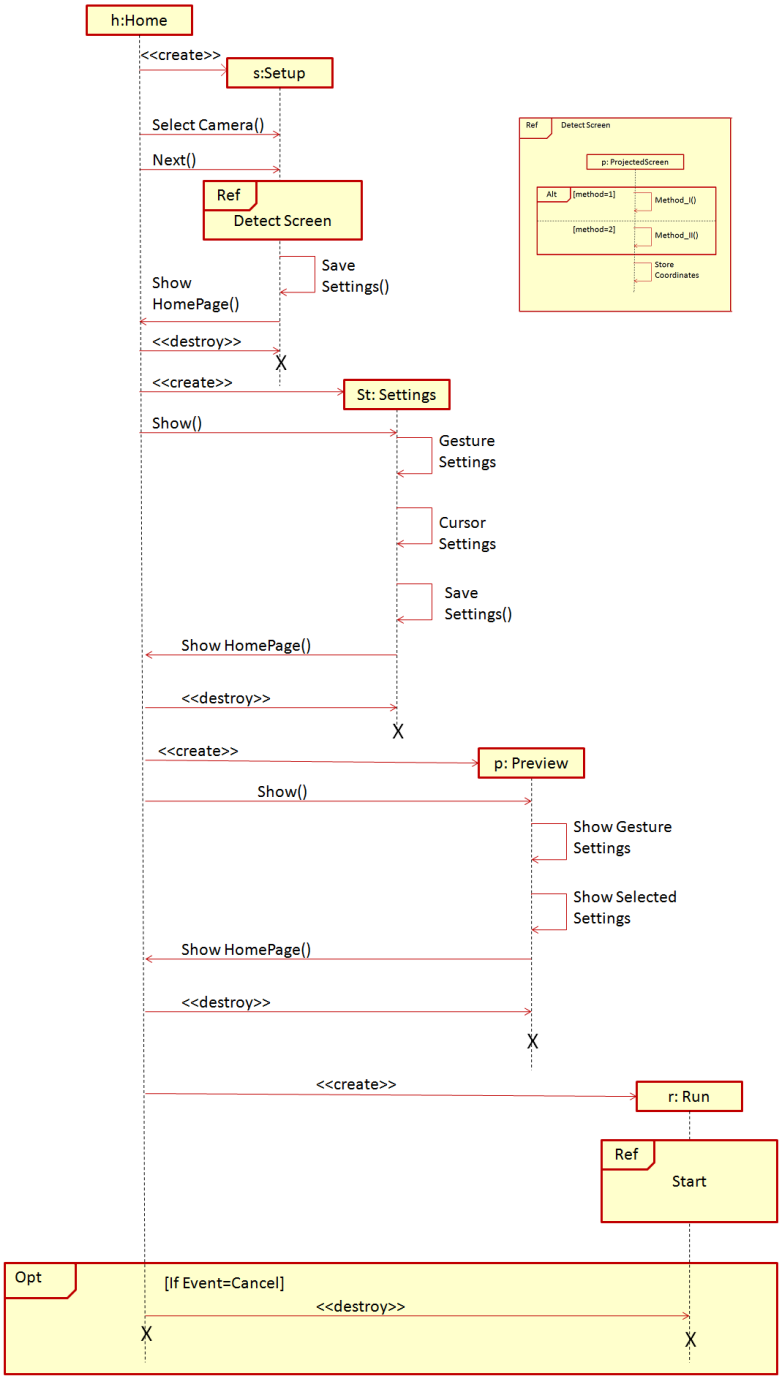


Figure 4.4: Sequence Diagram

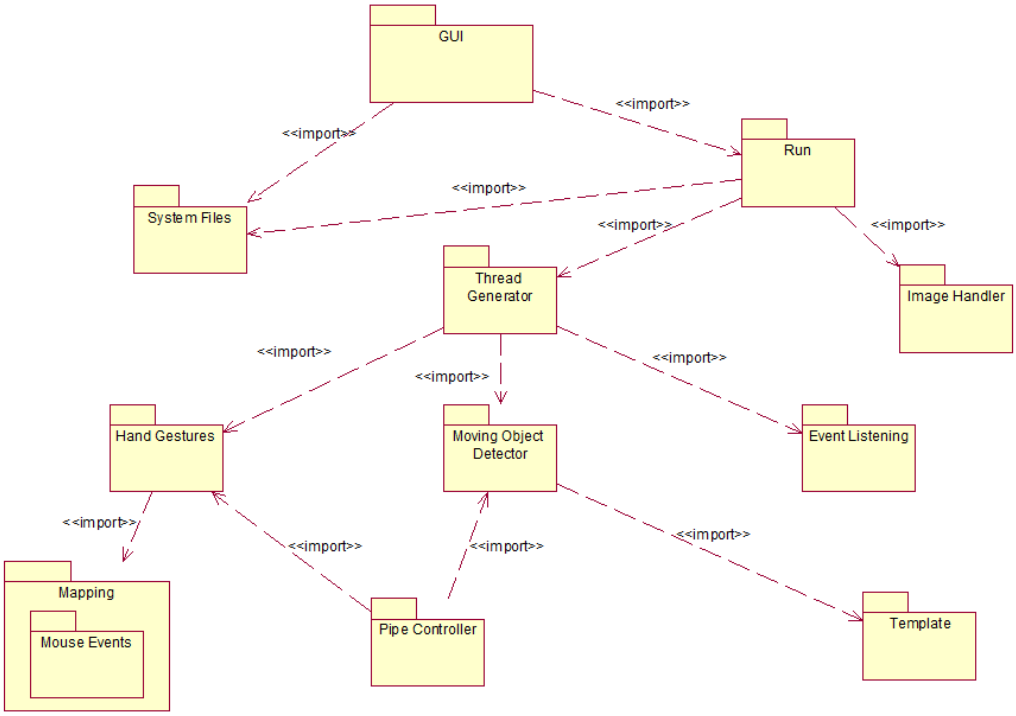


Figure 4.5: Package Diagram

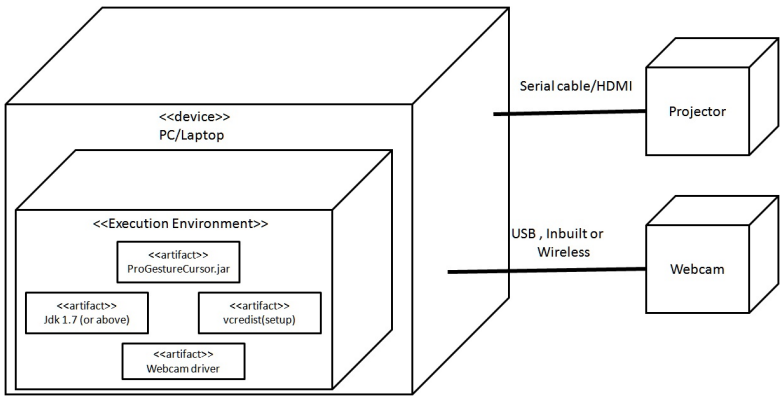


Figure 4.6: Deployment Diagram

Chapter 5

Technical Specification

Contents:

Chapter **5**

Technical Specification

5.1 TECHNOLOGIES USED

- **OpenCV framework :**

  OpenCV (Open Source Computer Vision Library) is a library of programming functions mainly aimed at real time computer vision. It is developed by Intel and now supported by Willow Garage.

  OpenCV is a free to use under Open Source BSD license. OpenCV mainly focuses on real-time image processing. It has C++, C and Python running on Windows, Linux, Android and Mac. The library has over 2500 optimized algorithms. OpenCV runs on Windows, Android, FreeBSD, OpenBSD, iOS, BlackBerry 10, Linux and OS X. OpenCV provides very good platform for developing robotics applications using image processing.

  One can also use Image processing toolkit from Matlab but OpenCV is much faster that Matlab image processing toolkit.

- **JavaCV Wrappers :**

  JavaCV is an image processing library. It provides wrappers for OpenCV. So that one can directly use OpenCV functions in Java. JavaCV also comes with hardware accelerated full-screen image display, easy-to-use methods to execute code in parallel on multiple cores (Parallel), user-friendly geometric and color calibration of cameras and projectors.

5.2 REFERENCES TO TECHNOLOGIES

- Links for OpenCV -
  - http://sourceforge.net/projects/opencvlibrary/files
  - http://opencv.willowgarage.com/wiki/InstallGuide
- Links for JavaCV -
  - http://code.google.com/p/javacv/downloads/list

Chapter 6

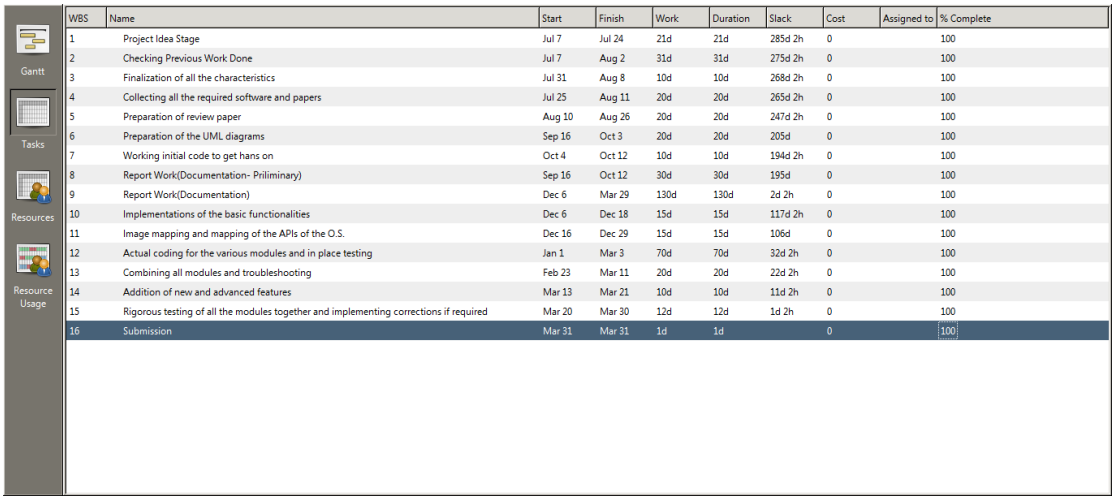
Project Management

Contents:

Chapter **6**

Project Management

6.1 PROJECT ESTIMATE



WBS	Name	Start	Finish	Work	Duration	Slack	Cost	Assigned to	% Complete
1	Project Idea Stage	Jul 7	Jul 24	21d	21d	285d 2h	0		100
2	Checking Previous Work Done	Jul 7	Aug 2	31d	31d	275d 2h	0		100
3	Finalization of all the characteristics	Jul 31	Aug 8	10d	10d	268d 2h	0		100
4	Collecting all the required software and papers	Jul 25	Aug 11	20d	20d	265d 2h	0		100
5	Preparation of review paper	Aug 10	Aug 26	20d	20d	247d 2h	0		100
6	Preparation of the UML diagrams	Sep 16	Oct 3	20d	20d	205d	0		100
7	Working initial code to get hans on	Oct 4	Oct 12	10d	10d	194d 2h	0		100
8	Report Work(Documentation- Priliminary)	Sep 16	Oct 12	30d	30d	195d	0		100
9	Report Work(Documentation)	Dec 6	Mar 29	130d	130d	2d 2h	0		100
10	Implementations of the basic functionalities	Dec 6	Dec 18	15d	15d	117d 2h	0		100
11	Image mapping and mapping of the APIs of the O.S.	Dec 16	Dec 29	15d	15d	106d	0		100
12	Actual coding for the various modules and in place testing	Jan 1	Mar 3	70d	70d	32d 2h	0		100
13	Combining all modules and troubleshooting	Feb 23	Mar 11	20d	20d	22d 2h	0		100
14	Addition of new and advanced features	Mar 13	Mar 21	10d	10d	11d 2h	0		100
15	Rigorous testing of all the modules together and implementing corrections if required	Mar 20	Mar 30	12d	12d	1d 2h	0		100
16	Submission	Mar 31	Mar 31	1d	1d		0		100

Figure 6.1: Project Estimate Planner

Sr.No.	Task	Technologies Used	Start Date
1	Project Plan - Domain Selection. - Project topic Selection.	-	3/7/2012
2	Literature Survey	IEEE explore	10/07/2012
3	Requirement Gathering - Technology Search. - Basic Functionality.	-	1/8/2012
4	Learning Technologies and Theory for Image Processing.	OpenCV JavaCV Net beans MS Visual Studio10 Visual Basic	5/8/2012
5	Implementation Of Basic Functionalities.	OpenCv JavaCv	15/8/2012

Continued on next page.

Table 6.1 – *From previous page.*

Sr.No.	Task	Technologies Used	Start Date
		Net beans	
6	IEEE paper search for Survey paper publication.	IEEE Explore Science direct Scopus	22/08/2012
7	Finalization of Software requirement specifications (SRS)	-	25/08/2012
8	Creation Of Preliminary Report	Latex	1/10/2012
9	Collection of Test Images considering various scenarios.	-	25/12/2012
10	Testing Of basic Functionalities	-	30/12/2012
11	Collection of test case and study of Haar classifier.(Hand And Head Images)	MS Visual Studio10	04/01/2013
12	Cursor Mapping - Mapping of cursor to hand position.	OpenCV JavaCV Net beans	03/02/2013
13	Finger-tip detection	OpenCV JavaCV Net beans	06/02/2013
14	Gesture Recognition. Based on number of fingers detected.	OpenCV JavaCV Net beans	08/02/2013
15	Testing of Haar classifier results. Analysis of modules for efficiency and optimization.	-	12/02/2013
16	Completion of Research Paper and publication	-	14/02/2013
17	Basic GUI creation	Java	28/02/2013
18	Integration of GUI with Code	Java	02/03/2013

Continued on next page.

Table 6.1 – *From previous page.*

Sr.No.	Task	Technologies Used	Start Date
19	Analysis And Optimization of modules - Finger-tip Detection. - Background Removal. - Use of Threading and pipelining.		06/03/2013
20	Final GUI creation and Integration.	Java	10/03/2013
21	Introducing Advanced features - Cursor Blocks. - Smooth Cursor Movement. - Manual Gesture Settings. - Shortcuts and PGC menu.		18/03/2013
22	Creation Of User Manual and Final Report	MS Word20/03/2013 Latex	

Table 6.1: Project Estimate

6.2 COST ESTIMATION

The Constructive Cost Model (COCOMO) is an algorithmic software cost estimation model developed by Barry W. Boehm. The model uses a basic regression formula with parameters that are derived from historical project data and current project characteristics.

Basic COCOMO computes software development effort (and cost) as a function of program size. Program size is expressed in estimated thousands of source lines of code (SLOC). COCOMO applies to three classes of software projects:

- **Organic projects** - "small" teams with "good" experience working with "less than rigid" requirements
- **Semi-detached projects** - "medium" teams with mixed experience working with a mix of rigid and less than rigid requirements

- **Embedded projects** - developed within a set of "tight" constraints. It is also combination of organic and semi-detached projects.

The basic COCOMO equations take the form
Effort Applied (E) = ab (SLOC) bb [man-months]
Development Time (D) = cb (Effort Applied) db [months]
People required (P) = Effort Applied / Development Time [count]
Where, SLOC is the estimated number of delivered lines (expressed in thousands) of code for project. The coefficients ab, bb, cb and db are given in the table 6.2.

Software Project	ab	bb	cb	Db
Organic	2.4	1.05	2.5	0.38
Semi-detached	3.0	1.12	2.5	0.32
Embedded	3.6	1.20	2.5	0.35

Table 6.2: SLOC coefficients

Basic COCOMO is good for quick estimate of software costs. However it does not account for differences in hardware constraints, personnel quality and experience, use of modern tools and techniques, and so on. We used TCL[1] and VDL[2] for SLOC calculation.

S.no	Package Name	TCL	VDL	NET LINES
1	GUI	7135	950	6185
2	Supportfiles.windowOption	500	50	450
3	System Files	275	50	225
4	Thread Generator	50	10	40
5	Event Listener	360	55	305
6	Hand Gesture	2138	95	2043
7	Image Handler	1306	50	1250
8	Mapping	180	10	170
9	Mouse Control	560	510	510
10	Moving Object	1460	60	1400
11	Pipeline handler	160	10	150
12	Run	190	10	180
13	Template	240	20	220
			TOTAL :	13128

Table 6.3: Lines Of Code

SLOC = 13.128(13.128 KLines)
Type of project: Organic

[1]Total Code Lines
[2]Variable Declaration Lines

- Effort applied (E):

  E = ab * (SLOC) * bb [Man-Months]

  E = 2.4 *13.128 * 1.05 Man-Months

  E = 33.0885 Man-Months

- Developement Time(D)

  D = cb * (Effort(E)) * db

  D = 2.5 * 33.0885 * 0.38

  D = 31.428 Months

- No. of Development Days = 6.5 months(approx.)

- People required (P)

  P = Effort/No. of Development Days (In Months)

  P = 33.0885/6.5

  P = 5.0905

  Therefore, estimated P for completion of this project is 5.

6.3 PROJECT SCHEDULE

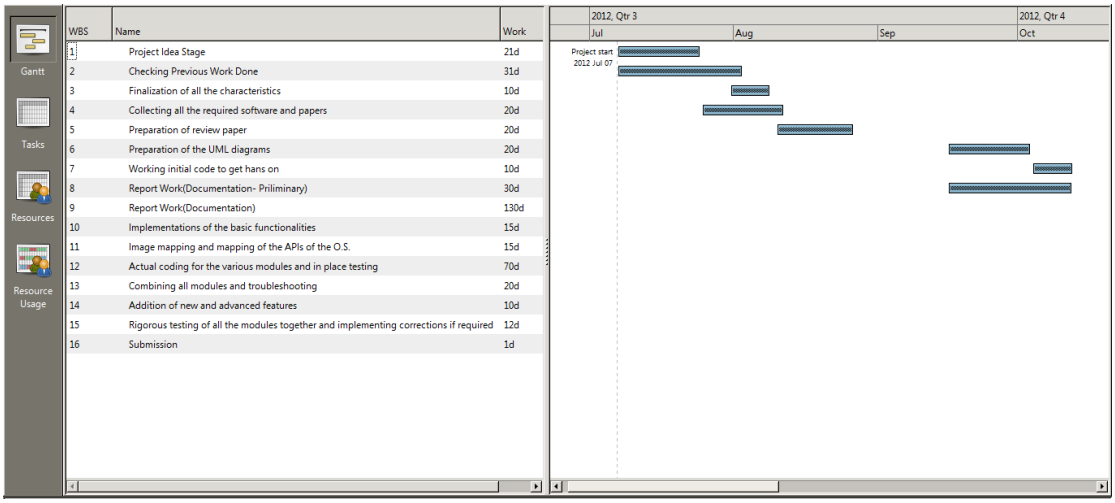
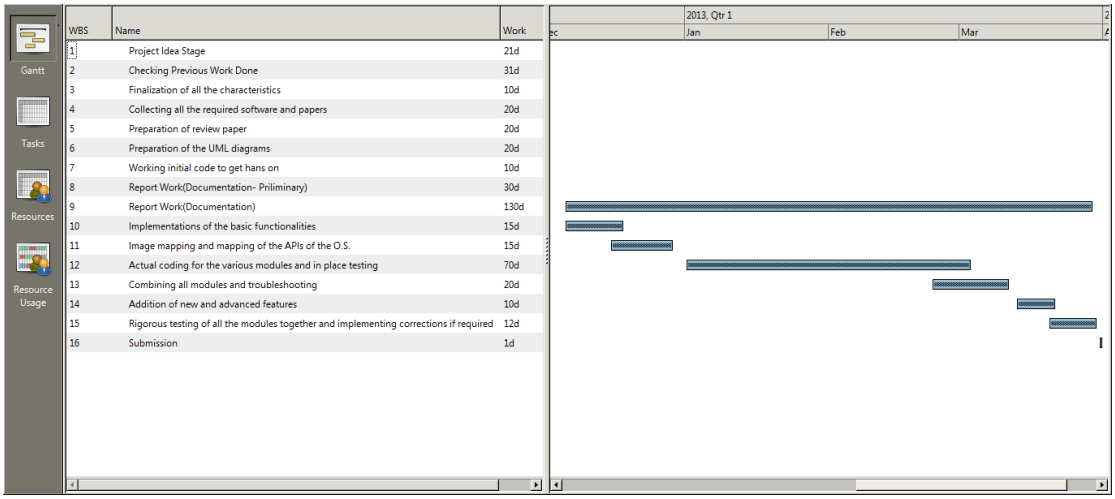


Figure 6.2: Project Schedule Phase I



WBS	Name	Work
1	Project Idea Stage	21d
2	Checking Previous Work Done	31d
3	Finalization of all the characteristics	10d
4	Collecting all the required software and papers	20d
5	Preparation of review paper	20d
6	Preparation of the UML diagrams	20d
7	Working initial code to get hans on	10d
8	Report Work(Documentation- Priliminary)	30d
9	Report Work(Documentation)	130d
10	Implementations of the basic functionalities	15d
11	Image mapping and mapping of the APIs of the O.S.	15d
12	Actual coding for the various modules and in place testing	70d
13	Combining all modules and troubleshooting	20d
14	Addition of new and advanced features	10d
15	Rigorous testing of all the modules together and implementing corrections if required	12d
16	Submission	1d

Figure 6.3: Project Schedule Phase II

6.4 TEAM STRUCTURE

The project team is organized in an ego-less team structure. Main features of ego-less team structure are :-

1. All team members assume responsibility for work done.
2. Team members cover all areas so that no one person or group monopolizes any one specific area of work;
3. Decisions are made by consensus.

Additional responsibilities on the team members -

1. Each person will work on multiple tasks throughout the development process in addition to the role assigned to him formally.
2. Due to the small size of team Project design, specification and requirements were discussed with the whole team.
3. One programmer will take the responsibility of a co-coordinator to ensure that the project is on schedule.

Figure 6.4: Team Members

1	Name	Anand Kulkarni
	Core Role	Programmer
	Additional Responsibility	GUI Design, Paper Publication
2	Name	Khursheed Ali
	Core Role	Programmer
	Additional Responsibility	Testing
3	Name	Himasnhu Tyagi
	Core Role	Programmer, Coordinator
	Additional Responsibility	GUI Design, Testing, Latex
4	Name	Snehal Govekar
	Core Role	Programmer
	Additional Responsibility	Documentation, Paper Publication, Testing

Table 6.4: Role of Members

Chapter 7

Software Implementation

Contents:

Chapter **7**

Software Implementation

7.1 INTRODUCTION

The main goal of this software is to make use of hand gestures thus making it cost effective. We have done modular programming to implement each of our 4 modules.

7.2 DATABASES

Hand features and Head features are stored in XML Files, while settings changed through GUI are stored in text files.

7.3 IMPORTANT MODULES AND ALGORITHMS/SAMPLE CODES

This system has been implemented in 4 main modules as follows:

1. Screen detection
2. Hand Recognition
3. Gesture Recognition
4. Calls to Windows API

7.3.1 Screen detection

We have devised two methods to find projected screen from the images captured by web camera. Method I: In this method, white and black images are projected through a projector and these images are captured by the web camera.Subtraction of white and black image[6] results into projected screen area.

```
detectscreen_Method_I() return void
{
      // Assuming white and black images do not contain any noise.

      TakeWhiteBlackSnap TWBS=new TakeWhiteBlackSnap(camera);
      bimg = TWBS.takeSnap(TWBS.black, 3);
      wimg=TWBS.takeSnap(TWBS.white,3);

      if flip=true then
        cvFlip(wimg, wimg,-1);
        cvFlip(bimg, bimg,-1;
```

```
        end if

        imageOperations iop=new imageOperations();
        fimg=iOp.Subtract(wimg, bimg);
        fimg=iOp.BW(iOp.GrayScale(fimg));
        findScrPos(fimg,wimg);
    }
```

Method II: In this method, white image is projected through a projector and captured by the web camera. This image is converted into HSV and by using specific threshold value, screen position is obtained.

```
detectscreen_Method_II() return void
{
   imageOperations iop=new imageOperations();
   TakeWhiteBlackSnap TWBS=new TakeWhiteBlackSnap(camera);
   wimg = TWBS.takeSnap(TWBS.white, 2);

   if flip=true then
       cvFlip(wimg, wimg,-1);
   End if
   int hL=0,hU=255;
   int sL=0,sU=255;
   int bL=30,bU=255;

   imageOperations iop=new imageOperations();
   img=iop.convertHSV(wimg);
   img=iop.thersholdImage(wimg, hL, sL, bL, hU, sU, bU);
   findScrPos(img,wimg);
}
```

7.3.2 Hand recognition algorithm

```
PipeLineStage1() return void
{
     PController PC=PController.createObject(difference.width(),
             difference.height(),difference.depth(),difference.nChannels());
     PC.startControler();
     while (ct.getRunningStatus())
     {
          ct.getImage(colourImageRoi,difference,greyImageAfterProcess,db);
          used=db.getDirtyBit();
          if colourImageRoi!=null && difference!=null &&
                    greyImageAfterProcess!=null  then

               iop.GrayScale(colourImageRoi, greyImage);
               hc=findContours(greyImageAfterProcess, colourImageRoi);
               chkAndFindHandCoorinate(hc, colourImageRoi,
                    greyImageAfterProcess,greyImage,htip);
```

```
                if used=false then
                     PC.passToNextStage(htip, hc, difference,colourImageRoi);
                     PC.passToNextStage(htip, hc, difference,colourImageRoi);
                end if
            end if
        }
        PC.stopControler();
    }
```

7.3.3 Gesture recognition algorithm

```
PipeLineStage2()  return void
{
        while (PC.getRunningStatus())
        {

                PC.getFromPreviousStage(htip, hc, difference,colourImageRoi);

                handGesture.detectHandGesture(htip,hc,difference);
                htip.put(handGesture.getHandTip());
                fingersCount=handGesture.getFingersCount();

                analyzeGesture.analyze(htip, fingersCount);
                htip.put(analyzeGesture.getActionPoint());
                action=analyzeGesture.getAction();

                cursorMap.map(htip,action,colourImageRoi);
        }
    }
}
```

7.3.4 Call to windows API

After mapping the hand position with cursor on computer screen, hand gesture is recognized and the Software calls Windows APIs to invoke various computer events.

```
leftClick() return void
{
    r.mousePress(InputEvent.BUTTON1_MASK);
    r.mouseRelease(InputEvent.BUTTON1_MASK);
}
middleClick() return void
{
    r.mousePress(InputEvent.BUTTON2_MASK);
    r.mouseRelease(InputEvent.BUTTON2_MASK);
}
```

```
rightClick() return void
{
    r.mousePress(InputEvent.BUTTON3_MASK);
    r.mouseRelease(InputEvent.BUTTON3_MASK);
}
leftPress()  return void
{
    r.mousePress(InputEvent.BUTTON1_MASK);
}
leftRelease() return void
{
    r.mouseRelease(InputEvent.BUTTON1_MASK);
}
```

Chapter 8

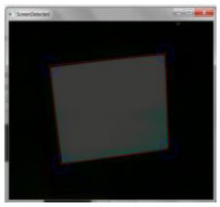
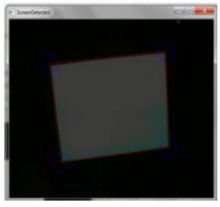
Software Testing

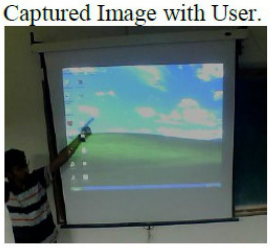
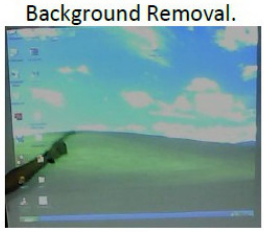
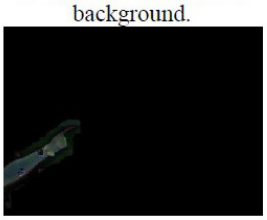
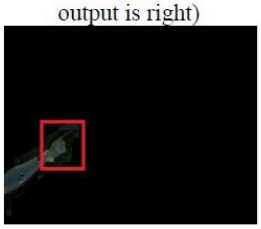
Contents:

Chapter 8

Software Testing

8.1 TEST CASES

Test-Case Id	Test Case/Input	Description	Expected output	Result
G1	User runs a system without performing setup first.	Setup must be performed at least once before starting the system.	Software must not run. Its should give some error message	Show message: - "No Settings Found."
G2	User changes gesture settings and closes a window without applying them.	User is allowed to change gesture settings as per his/her needs.	Default Settings must be applied	Default Settings or previous applied settings will be used.
G3	User Enters a preview button without performing a setup first.	User is allowed to preview settings before actually starting execution.	Should show the previous or default settings	Shows the previous and default settings
S1	Black Image  White Image 	These two images will be processed to get projected screen image. But Light difference between these two images should be almost same	Get co-ordinates of the projected screen.	Getting the Screen Coordinate 
S2	Projected Screen Co-ordinates 	Transform image into Perfect Rectangle.	Image must be transformed into a rectangle	Image after transformation 

Test-Case Id	Test Case/Input	Description	Expected output	Result
H1	Captured Image with User. 	Removing Unwanted Background.	Ouput image must not any unwanted background	Image After unwanted Background Removal. 
H2	Image with removed Background. 	Removing internal background	The internal background of the image must be deleted	Internal background removed. 
H3	Image without Internal background. 	Detection of the hand	Should get Hand coordinate	Locating hand position in the Image. (approx 90% output is right) 
H4	Detected hand image 	Moving of the mouse cursor to the hand location using the hand co-ordinates	Mapping of the mouse with hand location	Mouse should be where the hand is placed 

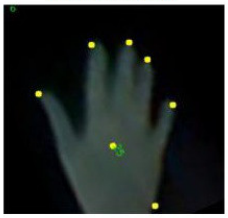
Test-Case Id	Test Case/Input	Description	Expected output	Result
H5	Image with hand 	Finding of the finger tips	All fingers tips must be detected and no extra fingers tips should be detected	Fingertips. Sometimes extra tips are also detected. 
H6	Determine Gesture 	Gesture Detection		Invoke Actions
		If gesture is for single click	Mouse should do single click	Working Fine
		If gesture is for double click	Mouse should do double click	Working Fine
		If gesture is for right click	Mouse should do right click	Working Fine
		If no gesture is found	No action should be done	Working Fine
C1	Black Image and white Image (For screen detection) (Automatic method I and II)	If detected screen is very close to the web camera	Show message: - "cannot able to work properly.	Showing message "Unable to detect screen"

Figure 8.1: Test Cases

Chapter 9

Critical Analysis Of Results

Chapter **9**

Critical Analysis of Results

We have implemented different algorithms to obtain better results at each stage during the development of project. Environmental conditions like brightness of projector, surrounding light intensity affects the results to a large extent. Taking these problems into consideration we have modified our algorithms. We have also analyzed the results after executing our software for particular period of time. The efficiency chart is shown in following figure.

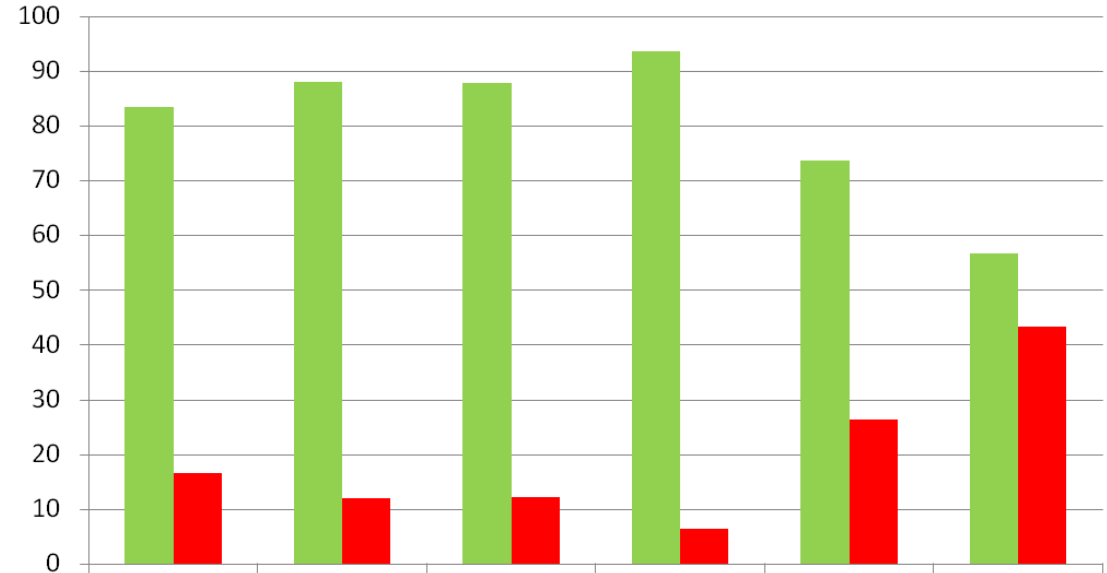


	Image Subtraction (Bright Enviroment)	HSB (Dark Enviroment)	Object Tracking using Thresholding	Object Tracking using Edge detection	Hand Detection horizontal check method	Hand Detection vertical check method
Pass	83.4	88	87.7	93.61	73.64	56.65
Fail	16.6	12	12.3	6.39	26.33	43.35

Figure 9.1: Graph showing efficiency of different algorithms

- *Image Subtraction (Screen Detection mode)*

  Image subtraction[6] method works efficiently in efficient light conditions. But sometimes camera introduces more noise as compared to black projected image, while capturing an image. This affects image subtraction result.

- *HSB bar (Screen Detection mode)*

Sometimes, the thresholding of the HSB image introduces noise in the output which decreases the efficiency of finding the correct screen coordinates.

- *Object Tracking[10] using Thresholding*

  Finding Threshold value dynamically is very difficult. So thresholding with the average value introduces noise (encircled part in fig. 9.2) when there is large disturbance in the moving object (hand).

Figure 9.2: Thresholding done with the average value introduces more noise

- *Object Tracking using Edge Detection*

  The noise introduced is relatively less as compared to Object tracking done using Thresholding method. Fig. 9.3 clearly reflects the large difference in noise as compared to thresholding method. Hence it is more efficient than Thresholding method.

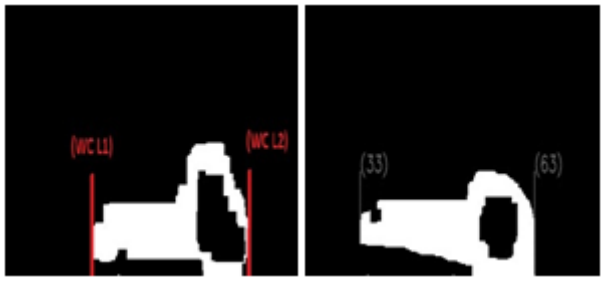
Figure 9.3: Object tracking done using Edge Detection introduces less noise

- *Hand Detection using Horizontal check method*

  This method uses Haar Classifier[11] and Fuzzy classification for finding the interested object and eliminating un-necessary objects (face, head etc.). Although Haar Classifier never gives 100% accuracy which affects the efficiency of this method.

- *Hand Detection using Vertical check method*

  Fig. 9.4 (A) and (B) was obtained by dilation and eroding the edge detected image. In this method, the two extreme points from left and right are found out. After that, from these points, the number of white pixels are counted up to the bottom of the image and the point having least WC (White Count) is taken as the tip of the hand. But in Fig. 9.4(A), WC L1 ¿ WC L2 (where L1 denotes Line 1 and L2 denotes L2) which is incorrect in our case.



(a) Incorrect Image as WC L1 ¿ WC L2 (b) Correct Image as WC L1 ¡ WC L2

Figure 9.4: Vertical check method

- *Face Detection using Haar Classifier*

  This algorithm fails when it detects something like head which is not actually face.

Chapter 10

Deployment And Maintenance

Contents:

Chapter 10

Deployment and Maintenance

10.1 INSTALLATION

- **Setup Package :**

  There are 2 types of setup depending on the Operating System:

  - x64 : 64 bit OS
  - x86 : 32 bit OS

- **Installation directory discription :**

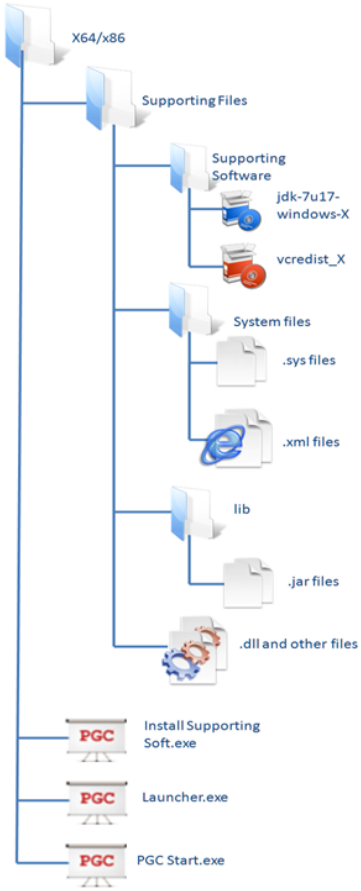


Figure 10.1: Installation directory discription

- x64/x86 folder contains one folder "Supporting Files" and three executable files:
  "Supporting Files" folder contains all the necessary files required for PGC application.
  "Install Supporting Soft.exe" can be used for installing supporting softwares.
  "Launcher.exe" , "PGC START.exe" (either) can be used for launching PGC.
- Supporting Files Folder:
  Its has 3 folders in it and some .dll files.
  dll files : OpenCv dll.s required for running PGC.
  "lib" folder : contains all the .jar's files (including JavaCv's)
  "Supporting software" folder : contains .exe of all supporting softwares.
  "System Files" folder : contains .sys and .xml file used by PGC at runtime.

- **Installation steps:**

1. Copy folder x86 OR x64 (depending upon the OS),into the directory where you want to install PGC.
2. Open the copied folder.
3. Double Click "Install Supporting Soft.exe". It will then automatically install java and vcredist(May take few mins to install).
4. Double Click "Lanucher.exe" to launch PGC. But in case PGC fail to launch, then set the "path" of java directory(where you have installed it) into the "Environment variable". example :

   ```
   "C:\Program Files\Java\jdk1.7.0_03"
   ```

   Then run "PGC START.exe" to launch PGC.
5. Create a shortcut for either of the exe.s on desktop to launch PGC next time.

10.2 UN-INSTALLATION

Un-Installation of PGC is very simple. Just delete the folder copied in the installtion procedure. User can also remove Java and vcredist from Control Panel if they are not required.

If a user wants to keep the saved settings of PGC, before deleting the above folder, just copy .sys files from

```
"\Supporting Files\System Files\"
```

and replace them when ever new copy of PGC is installed.

10.3 USER HELP

It is essential for the user to know how the software works. Hence User help is provided in the software as well as a user manual is provided to guide him through the entire process.

Help button is provided at top right corner of every window to guide the user if he gets stuck at any point. After clicking on '?' button, help is provided regarding functioning of every software setting.

Figure 10.2: Setup Settings help

Chapter 11

Future Scope And Conclusion

Contents:

Chapter **11**

Future Scope And Conclusion

11.1 FUTURE SCOPE

This project eliminates the use of mouse and keyboard during presentations and make them all more interactive and present a wide screen computer to control. The project scope mentioned above can be extended to various fields. Controlling projected computer system using hand gesture can find its application in

- Presenting lectures
- Conducting seminars
- Business presentations
- Statistical reports

in more effective and efficient way by merely standing beside the projected screen.

The scope of this project can be extended by creating profile of every user (e.g. faculty members) and loading respective profile using face recognition when user stands in front of web camera. Initially, user can save any predefined his/her hand gestures instead of using default gestures. This saves time of changing the gesture settings every time that user conducts lecture or gives presentation.

11.2 CONCLUSION

The main purpose of this project was to get rid of old traditional dependence on keyboard, mouse or other hardware while presenting lectures, giving seminars etc. This project aims at reducing the overhead of additional hardware required while interacting with the computer.

- This software can be used to give presentations. One can change the slides, play custom animations by merely standing beside the projected screen and making corresponding gestures.
- Using this system, user can control the computer system through gestures only, by performing click or double click on the icons, scrolling windows and much more.

- The system enables a user to be more creative and expressive, as, while giving presentations the system simulates a touch screen (projection).
- Compared to other similar solutions, this system is cheaper and general purpose.

Bibliography

[1] http://qph.cf.quoracdn.net/main-qimg 17601437f077288f37a5d99eaea5372. Kinect technology image.

[2] E. Foxlin. Motion tracking requirements and technologies. In *Handbook of Virtual Environment Technology*, page 163210, 2002.

[3] H. Stern J. P. Wachs, M. Klsch and Y. Edan. Vision-based hand-gesture applications. In *Communications of the ACM*, page 54:6071, 2011.

[4] Shihwen Hsiao and Yi cheng Tsao. Color- banded hand tracking system based on fuzzy logic concept. 2011.

[5] Heng Du and TszHang To. Hand gesture recognition using kinect. December 15, 2011.

[6] [Online]. Available: http://docs.opencv.org/search.html?q=cvabsdiff. In *Image Subtraction*.

[7] [Online]. Available: http://opencv.willowgarage.com/documentation/geometric-image-transformations.html. In *Image Transformation*.

[8] [Online]. Available: http://mateuszstankiewicz.eu/ ?p=189. In *Background Subtractor MOG2*.

[9] [Online]. http://fivedots.coe.psu.ac.th / ad/jg/nui055/. In *Gesture detection*.

[10] [Online]. Available: http://opencv.willowgarage.com /documentation/c/ imgproc-motion-analysis-and-object-tracking.htm. In *Object Tracking*.

[11] [Online]. Available: http://note.sonots.com/scisoftware /haartraining.html. In *Harr Training*.

Appendix **A**

Jargons

- Gesture recognition :

  Gesture recognition is a topic in computer science and language technology with the goal of interpreting human gestures via mathematical algorithms. Gestures can originate from any bodily motion or state but commonly originate from the face or hand

- Image Processing :

  In imaging science, image processing is any form of signal processing for which the input is an image, such as a photograph or video frame; the output of image processing may be either an image or a set of characteristics or parameters related to the image. Most image-processing techniques involve treating the image as a two-dimensional signal and applying standard signal-processing techniques to it.

- Fuzzy Logic :

  Fuzzy logic is a form of many-valued logic or probabilistic logic; it deals with reasoning that is approximate rather than fixed and exact. In contrast with traditional logic they can have varying values, where binary sets have two-valued logic, true or false, fuzzy logic variables may have a truth value that ranges in degree between 0 and 1.

- Kinect :

  Kinect is a motion sensing input device by Microsoft for the Xbox 360 video game console and Windows PCs. Based around a webcam-style add-on peripheral for the Xbox 360 console, it enables users to control and interact with the Xbox 360 without the need to touch a game controller, through a natural user interface using gestures and spoken commands.

- Web Camera :

  A webcam is a video camera that feeds its images in real time to a computer or computer network, often via USB, ethernet, or Wi-Fi.

- Long Throw Projector :

  The projector and the screen have influence on each other. Throw distance refers to the distance between the projector and the screen.

- Windows APIs :

  The Windows API, informally WinAPI, is Microsoft's core set of application programming interfaces (APIs) available in the Microsoft Windows operating systems. The name Windows API collectively refers to a number of different platform implementations that are often referred to by their own names (for example, Win32 API); see the versions section. Almost all Windows programs interact with the Windows API; on the Windows NT line of operating systems, a small number (such as programs started early in the Windows startup process) use the Native API.

- Data Flow Diagrams :

  A data flow diagram (DFD) is a graphical representation of the "flow" of data through an information system. It differs from the flowchart as it shows the data flow instead of the control flow of the program. A data flow diagram can also be used for the visualization of data processing (structured design).

- UML Diagrams :

  Unified Modeling Language(UML) is a standardized general-purpose modeling language in the object-oriented software engineering. UML combines techniques from data modeling (entity relationship diagrams), business modeling (work flows), object modeling, and component modeling. It can be used with all processes, throughout the software development life cycle, and across different implementation technologies.

Appendix **B**

Achievements

- BEST PROJECT OF THE YEAR, Department Of Computer, VIIT.
- 1st prize in CSI national conference.
- 1st prize in CSI national project competition.
- 1st prize in Convene 2k13 national project Competition.
- Paper published in International journal.
- Paper published in CSI Journal.
- Papers accepted in 4 International and 5 national Journals.

Appendix **C**

Published Papers

- **Survey Paper :**

  Published in -

  International Journal of Advances in Management, Technology And Engineering Sciences (IJAMTES)

  ISSN : 2249 - 7455

  Vol. I, *Issue 12(III)*, September 2012

- **Research Paper :**

  Published in -

  National Conference and Project Competition for Students in Computer Engineering And Information Technology(NCPCI)

  Vol. I, *Issue 1*, March 2013

Controlling projected computer system using hand gesture recognition technique

Anand L. Kulkarni, Himanshu Tyagi, Khursheed Ali, Snehal N. Govekar, Prof. Y. K. Sharma

Department of Computer Science
Vishwakarma Institute of Information Technology, Pune, India
Email ID : **anandkulkarni67@gmail.com, himanshutyagi1990@yahoo.com**
Contact : +91 98 9064 3268, +91 86 0504 8160

***Abstract* :**

Hand gesture recognition based man-machine interface is being developed vigorously in recent years because of the great emphasis on easing the Human-Computer-Interaction. Currently, electro-mechanical or magnetic sensing devices such as data gloves are the most effective tools for capturing hand gestures. Also vision-based hand gesture recognition is a promising alternative to them because of its potential to provide more natural and non-contact interaction. In this paper, we propose a mechanism to simulate a "touch screen projector". This will make use of hand gesture recognition technique to control the computer system with various commands. Different hand gestures will be used to control computer (specifically cursor) by standing in front of projector. These events may include single click, double click, scrolling, dragging etc. Also, cursor on the screen will move according to hand movement.

Keywords : Hand Gesture Recognition, Human-Computer-Interaction, Vision based.

I. INTRODUCTION

With the need of employing natural communication between humans and machines there has been a great emphasis on Human-Computer-Interaction (HCI) lately. Hand being the most effective general-purpose tool because of its dexterity, adopting hand gesture as an interface in HCI will not only allow the deployment of a wide range of applications in sophisticated computing environments such as virtual reality systems and interactive gaming platforms, but also benefit our daily tasks by controlling interfaces via gestures[2]. Electro-mechanical or magnetic sensing devices such as data gloves[1] are the most effective tools for capturing hand gestures. These tools have sensors attached to them(e.g. glove) that converts finger movements into electrical signals to determine the hand gesture. However, they have several drawbacks 1. for casual use, they are very expensive 2. with wires etc attached, they hinder naturalness of hand gesture, and 3. they require complex calibration and setup procedures to obtain precise measurements. Vision-based hand gesture recognition serves as a promising alternative to them. They provide more natural, unencumbered, non-contact interaction.

In this paper we propose to use the same, vision based hand gesture recognition system, to provide HCI through projected computer screen. Today when we need to use projector, its tedious to interact with computer system at the same time. One needs help of another person to control the computer while he/she uses the projector on the dais. Solutions to specific problems(Presentations, Gaming etc.) are available in this context but no exact solution is present for it as a whole. Also the present problem specific solutions have their limitations and are not cost effective as well. Our paper proposes solution to control the entire computer which solves every other small problems talked about earlier.

II. RELATED WORK

First, we give a brief overview of related vision-based projected computer control approaches :

I. Colour band based hand tracking system:

One of the existing technologies present is Colour band Hand Tracking System (CbHTS)[3]. It is slightly different than the traditional mouse. User's fingers will have different coloured bands on them. CbHTS scans specific zone for colour bands, and the dimension of scan zone is determined by applying fuzzy logic on cursor speed of previous image. One of the user's fingertip is treated as the mouse cursor, and single click is defined as the contact of two fingers. Double click is defined as continuous two single clicks. CbHTS concludes the centre of each finger band ,it proceeds to the judgment of clicks by analyzing the difference between time period of the consecutive clicks. But in this method, colours of bands and background plays a vital role. Thus, it requires a camera and an algorithm which can differentiate band and background color precisely. It requires that the surrounding colors should not be same as that of the color bands and the band colours should have noticeable difference in their RGB levels. Also only predefined colors can be used this type of tracking system.

II.. Computer control using LED stick:

This is another technology that is used for gesture recognition in front of the projector. A stick is used in this method and the stick has a LED attached to its tip. When it is moved in front of the projected image, the camera detects glowing LED and moves the cursor to the respective location on the screen. The stick has two buttons, one for left-click and another for right-click. When these buttons are pressed , the respective actions are performed at the location pointed by the stick. This technique is expensive to implement, since it requires number of hardware components. Also it is not flexible to use a stick every time. Color of the LED matters in this vision based tracking system as in some cases its color may not get detected because of the background's color.

III. Kinect based hand tracking :

Kinect[4] is 3-D camera, with high depth sensing capability, it has given birth to various exciting systems in human-machine interaction. One potential application is in the “playing games with hand motion” environment where a person can play games on computer without physically touching consoles etc. With Kinect, processing is fast and it gives the higher efficiency. But affording it for the general purpose is very costly and can't be used for normal activities. Also the size of it cannot be accommodated in the desired location/manner.

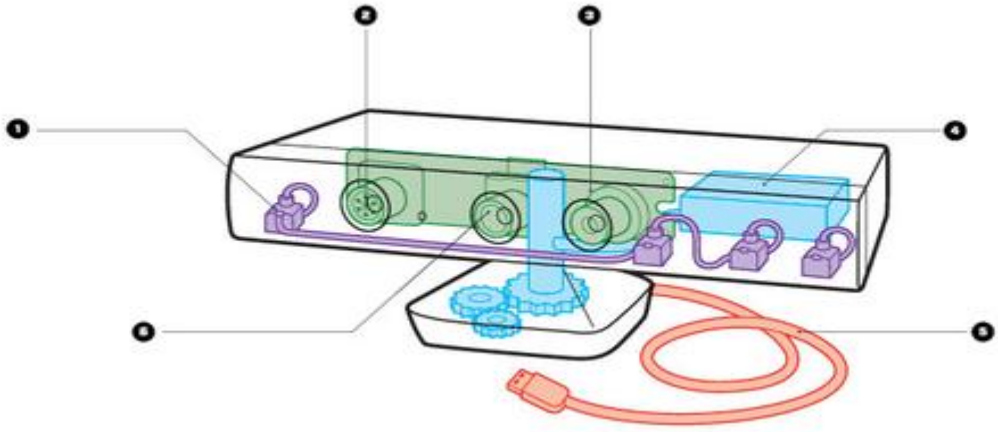


Fig 1. Kinect technology[5]

1. Microphone array - Four mics pinpoint where voices or sounds are coming from while filtering out background noise.

2. IR emitter - Projects a pattern of infrared light into a room. As the light hits a surface ,pattern becomes distorted, and the distortion is read by the depth camera.

3. Depth camera - Analyzes IR patterns to build a 3-D map of the room and all objects and people within it.

4. Tilt monitor - Automatically adjusts based on the object in front of it. If you're tall, it tilts the box up. If you're short, it knows to angle down.

5. USB cable - Transmits data to the Xbox via an unencrypted feed, which makes it relatively easy to use the kinect with other devices.

6. Colour camera - Like a webcam, this captures a video image. The kinect uses that information to get details about objects and people in the room.

III. PROPOSED TECHNIQUE

The proposed application will be able to handle all the computer events using hand gestures only. The user will be able to control all the computer events through hand gestures only. The application will be cost effective since it will require only a single webcam for capturing the runtime changes in front of the projected screen. Camera will be mounted on or near the projector so as to capture complete image as projected by the projector[Fig. 2]. The web camera will capture the images at regular intervals only when there is motion in front of it. These images will be compared with the actual desktop image at respective instant of time. By using image subtraction algorithm these images will be analyzed. The gesture made by the user in front of the projector screen will be extracted, using different image processing algorithms. The gesture made by the user will be identified from the extracted information and the actions(single click, double click etc) corresponding to that particular hand gesture will be performed. This system can be used to give presentations using hand gestures. One can change the slide, click or double click the icons, folders, scroll the window by merely standing beside the projected screen. Every computer event has a unique hand gesture to invoke its functionality.

Figure 2 shows the positioning of various components in the project. Projection defines the projected region on the screen. While in Figure 3 there are hand gestures which act as obstruction and camera detects them.

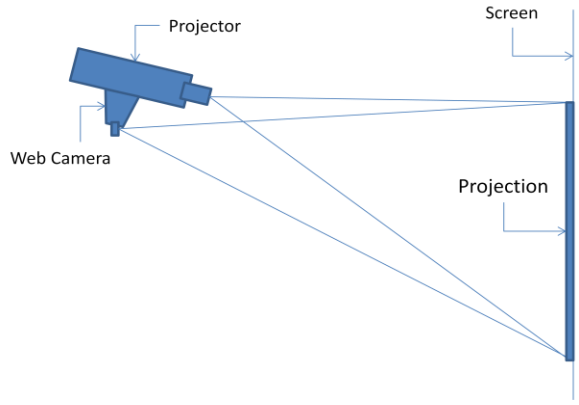


Fig.2: Computer Projection Without User

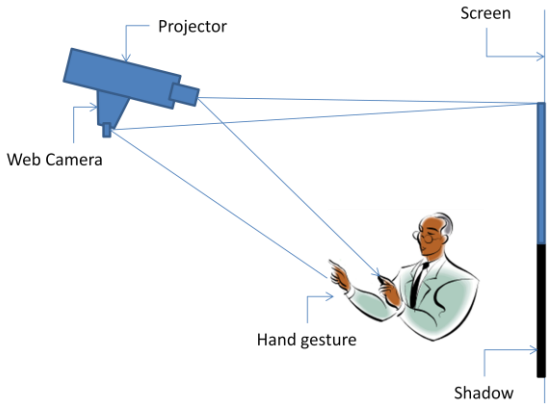


Fig 3: Computer Projection With a User

Few additional features in this would be minimizing, maximizing, closing the window using respective hand gestures. It also eliminates the need for remote to control various activities during the presentation. This system does not recognize any colors and thus foreground (person operating) and background (image projected) does not matter. But as this project work on web-cam, the efficiency and correctness of the project depends on the resolution and quality of the camera. Distance between camera and the projected screen matters a lot. As the distance increases above the threshold value the efficiency of detection keeps on decreasing because the mapping starts to become difficult.

IV. CONCLUSION

This paper discussed about various vision based HCI systems with their pros and cons. Also a novel idea of controlling projected computer was proposed and discussed in a brief manner in the paper. The proposed idea frees the user from using any data gloves or sticking any colored strip on their fingers to control the computer. The basic idea behind the system is to recognize user's hand movement and gestures using image processing. This facilitates the user to freely interact with the computer system.

V. REFERENCES

[1] E. Foxlin. Motion tracking requirements and technologies. Handbook of Virtual Environment Technology, pages 163–210, 2002.

[2] J. P. Wachs, M. Klsch, H. Stern, and Y. Edan. Vision-based hand-gesture applications, Communications of the ACM, 54:60–71, 2011.

[3] Shihwen Hsiao, Yi-cheng Tsao, Color- banded hand tracking system based on fuzzy logic concept.

[4] Heng Du, TszHang To, Hand gesture recognition using kinect.

[5] Kinect Technology Image , http://qph.cf.quoracdn.net/main-qimg-17601437f077288f37a5d99eaea537

Controlling projected computer system using hand gesture recognition

Anand L. Kulkarni, Himanshu Tyagi, Khursheed Ali, Snehal N. Govekar, Prof. Y. K. Sharma
Department of Computer Science
Vishwakarma Institute of Information Technology, Pune, India
anandkulkarni67@gmail.com, tyagihimanshu@outlook.com

***Abstract* — Hand gesture recognition based Human-machine interaction interface is being developed vigorously in recent years because of the great emphasis on easing the Human Computer Interaction. This paper gives an implementation of an application which can be used to control the projected computer screen using hand gestures. The purpose of this project is to provide a mechanism to simulate a "Touch Screen Projector". Different hand gestures will be used to control various computer events (specifically cursor based events) by standing in front of projector. The events may include single click, double click, scrolling etc. The setup of this project includes mounting a web camera near the projector. The web camera will be capturing real time projected images. These images will be processed to recognize hand gestures and corresponding events will be invoked.**

Keywords: Hand Gesture Recognition, Human Computer Interaction, Image Processing, Projected Computer.

I. INTRODUCTION

Lately, the importance of man-machine interface has been recognized by the IT industry and emphasis has been given on the Human-Computer Interaction (HCI). One of the ways of achieving the natural communication between humans and machine is through Gestures. The existing development in this field includes electro-mechanical or magnetic sensing devices such as data gloves [1] which are some of the effective tools for capturing hand gestures. Vision-based hand gesture recognition [2] is a promising alternative to these techniques (mentioned above) because of its potential to provide more natural and non-contact interaction.

In today's world, business presentations, seminars have become a daily routine in IT industry. In case of presentations, seminars etc., it is tedious to use other devices(lasers etc) or computer peripherals(keyboard, mouse etc) while giving presentations. The traditional method of projecting the computer screen and controlling the computer events by keyboard and mouse consumes time as the presenter has to walk all the way to the computer and then do the changes. Another option for this would be to take help from an assistant which cannot be possible every time.

Few of the existing technologies sharing our project base are Color band based Hand Tracking System(CbHTS)[3], Kinect based Hand Tracking System[4]. Although many gesture recognition systems are available in the market, most of them are problem-specific and are not cost-effective as well.

So, in this paper we present a solution to control entire computer system, projected on a plane surface, without using any accessories such as data gloves, lasers or LED sticks etc. It will make use of hand gesture recognition techniques to control the computer system. By using various hand gestures, user can interact with our system.

II. PROPOSED TECHNIQUE

The proposed system will be able to handle computer events using hand gestures only. The application will be cost effective since it will require a normal web camera for capturing the runtime changes in front of the projected screen. Camera will be placed in such a position, that it captures complete projected image. When the user will be interacting with the computer system, using hand gestures, the web camera will capture images at regular intervals, only when it detects any movement. The projected images captured by the web camera will be processed using various Image Processing algorithms to recognize the hand gestures. Finally, the mapping of hand gesture and mouse movements will be performed to invoke computer event(s) (single click, double click etc.) at the intended screen position.

III. ALGORITHM

1) *Projected Screen Detection.*
   a. When the application starts, it first captures and detects the projected screen using web camera.
   b. It then transforms the image into a perfect rectangle, if necessary.

2) *Gesture Detection.*
   a. Hand is detected in the captured image.
   b. After that, the application recognizes hand gestures, if any.

3) *Invoking an event.*
   a. Intended position for the event is mapped on desktop.

b. According to the recognized hand gesture, corresponding computer event is invoked.

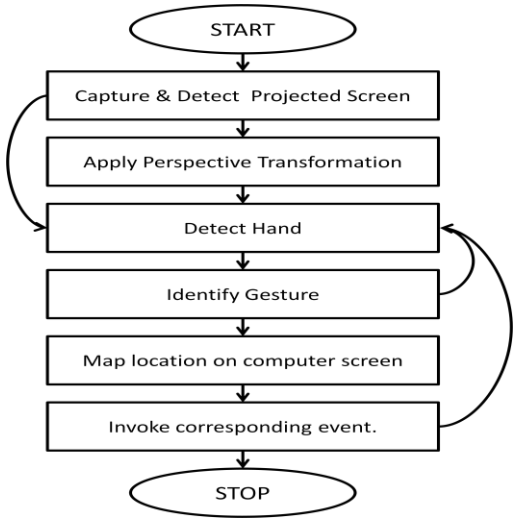


Fig. 1 System flowchart.

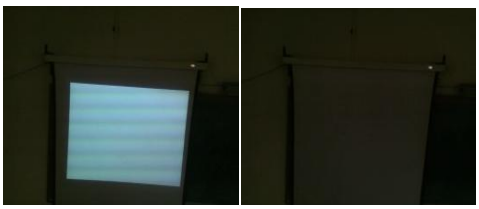
IV. WORKING

A. Projected Screen Detection

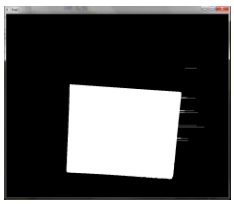
It is necessary to detect the projected screen for the purpose of mapping mouse movements and hand gestures. It can be achieved using any of the following methods:

1) Method 1: Black and White projected image subtraction

In this method, first a completely white(Fig 2.A) and black(Fig 2.B) images are projected and captured. Then both the images are subtracted[5] to obtain projected screen region(Fig 2.C).



(A) (B)



(C)

Fig. 2 (A) White projected screen
(B) Black projected screen
(C) Subtraction of (A) and (B)

For finding corners of the screen, two different methods can be used.

a. By finding the contour of the subtracted image.
b. By finding the intersection points of all the edges from maximum white cluster in the subtracted image.

2) Method 2: Using only White projected image

a. Convert the RGB image into HSB(a variation of the HSV color space image, where H: Hue, S: Saturation, B: Brightness).
b. Using appropriate Threshold values of H, S and B, filter the HSB image.
c. Find contours of the filtered image.
d. Find convex hull of the biggest contour image which will give us coordinates of the projected screen as shown in Fig. 3.

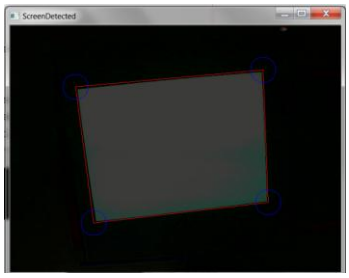


Fig. 3 Corner Coordinates of the projected screen

B. Perspective Transformation

Using screen coordinates found previously, perspective transformation[6] is applied on the real time images, captured by the web camera. This results into a perfect rectangle as shown in Fig. 4.

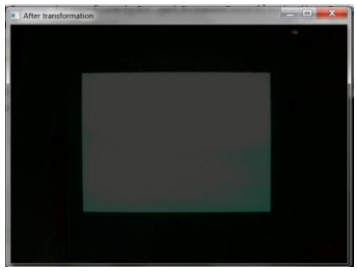


Fig. 4 Projected screen obtained after applying Perspective Transformation to get a perfect rectangle

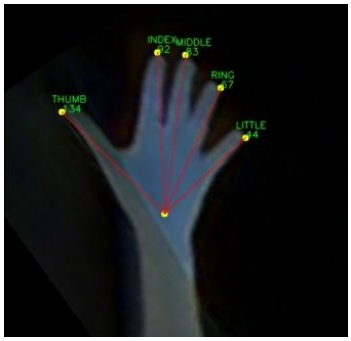
C. Hand Detection

Hand can be detected by tracking the moving object (human body) in front of the projected screen. Moving object tracking is done by dividing the image into two parts: first is any moving object and the other is its

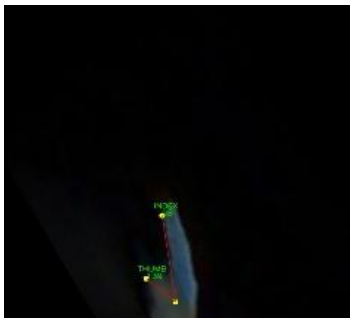
background[7]. Then, hand is detected using Haar Classifiers[8] generated by training.

D. *Gesture Recognition*

For identifying the gestures, we need to find finger tips [Fig. 5.a]. Then, gestures are recognized[9] using the positions and angles made by the finger tips. For example, index finger and thumb is used for moving the cursor as shown in Fig. 5.b.



a. Gestures are recognized from finger tips



b. Thumb and index finger is used to move the cursor

Fig. 5 Gesture recognition.

E. Invoking Computer Event

The final stage is to invoke a computer event corresponding to the hand gesture. When we get the hand gesture, mapping is done between the hand position (which acts as a cursor) and the actual screen coordinates. Accordingly, the corresponding computer event like single click, double click or drag etc. is invoked at the hand position.

V. ANALYSIS OF ALGORITHMS

We have used different algorithms in the above mentioned five stages to achieve the best possible results. Some of which were more efficient than the other methods or yielded better results. Following is the analysis of the algorithms used:

A. The Image Subtraction bar (Screen Detection mode)

In bright environment, image subtraction method works efficiently. But sometimes while taking the image of white projected screen, the camera introduces more noise in the background as compared to black projected screen. This causes a failure in this method.

B. HSB bar (Screen Detection mode)

When it is not too dark, thresholding of HSB image introduces noise in the output which decreases efficiency of finding the correct screen coordinates.

C. Object Tracking using Thresholding

Finding Threshold value dynamically is very difficult. So thresholding with the average value introduces noise (encircled part of Fig. 6) when there is too much disturbance in the moving object (hand).

Fig. 6 Thresholding done with the average value introduces more noise

D. Object Tracking using Edge Detection

The noise introduced is relatively less as compared to Object tracking[10] done using Thresholding method. Fig. 7 clearly reflects the large difference in noise as compared to thresholding method. Hence it is more efficient. Here, Canny Edge Detection is used.

Fig. 7 Object tracking done using Edge Detection introduces less noise

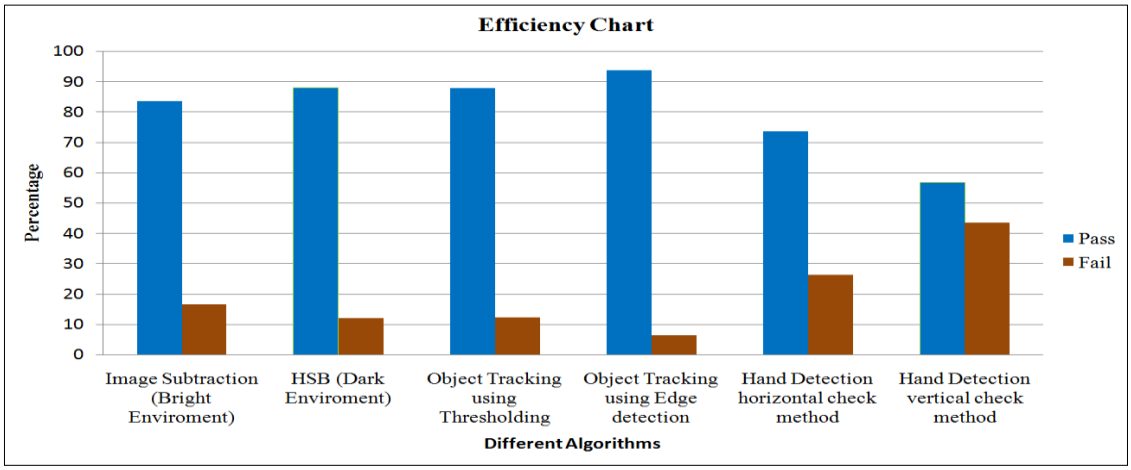


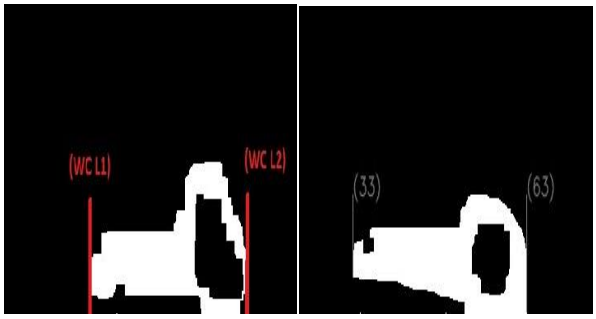
Fig. 8 Graph showing efficiency of different algorithms

E. Hand Detection using Horizontal check method

This method uses Haar Classifier for finding the object of interest (hand, face, head etc.). Further it is used for differentiating between hand and other objects.

F. Hand Detection using Vertical check method

Fig. 9 (A) and (B) was obtained by dilation and eroding the edge detected image. In this method, the two extreme points from left and right are found out. And then to find out on which side the finger is, number of pixels vertically on both sides are counted. Side with lesser pixels means a pointing finger.



(A) (B)

Fig. 9 (A) Incorrect Image as WC L1 > WC L2
(B) Correct Image as WC L1 < WC L2

Fig. 8 explains the efficiency graph of different algorithms used in each of the five stages mentioned before. The bar graph shows passed and failed percentage of the test cases, for each algorithm used.

VI. OPERATING ENVIRONMENT

Operating requirements for the system are :

A. User Interfaces
Frontend software : Developed GUI Application software.
System type : Stand-alone system.

B. Hardware Interfaces
RAM : 1 GB (min.)
Web Camera : 8 MP (min.)
Other : Projector (long throw)

C. Software Interfaces
Operating System : WindowsXP/7/8, Linux
Framework : JavaCV (OpenCV wrapper for Java)
Programming Language : Java (JDK6.1)
Database : XML,File system.
Others : Windows APIs

VII. ADVANTAGES

1) This software can be used to give presentations. One can change the slides, play custom animations by merely standing beside the projected screen and making corresponding gestures.
2) Using this system, user can control the computer system through gestures only, by performing click or double click on the icons, scrolling windows and much more.
3) The system enables a user to be more creative and expressive, as, while presentations the system simulates a touch screen (projection).
4) Compared to other similar solutions, this system is cheaper and general purpose.

VIII. CONCLUSION

Using image processing open source libraries, very basic web camera and Java code this system was implemented successfully. The application designed for the system was able to track hand and move the operating

system cursor. It also detected hand gestures and performed desired events.

This system is another solution for HCI wherein interaction is smooth and intuitive. Using above mentioned steps and algorithms the system gives desirous results and acceptable accuracy and speed.

REFERENCES

[1] E. Foxlin. Motion tracking requirements and technologies. Handbook of Virtual Environment Technology, pages 163–210, 2002.

[2] J. P. Wachs, M. Klsch, H. Stern, and Y. Edan. Vision-based hand-gesture applications, Communications of the ACM, 54:60–71, 2011.

[3] Shihwen Hsiao, Yi-cheng Tsao, Color- banded hand tracking system based on fuzzy logic concept.

[4] Heng Du, TszHang To, Hand gesture recognition using kinect.

[5] Image Subtraction : [Online]. Available: http://docs.opencv.org/search.html?q=cvAbsDiff

[6] Image Transformation: http://opencv.willowgarage.com/documentation/geometric_image_transformations.html

[7] BackgroundSubtractorMOG2: http://mateuszstankiewicz.eu/?p=189

[8] Harr Training: [Online]. Available: http://note.sonots.com/SciSoftware/haartraining.html

[9] Gestures detection: http://fivedots.coe.psu.ac.th/~ad/jg/nui055/

[10] Object Tracking: [Online]. Available: http://opencv.willowgarage.com/documentation/c/imgproc_motion_analysis_and_object_tracking.htm