MULTI TOUCH SURFACE SYSTEM

<u>Submitted By:</u>

Wajdan Zahid

(01-233072-013)

Umair Javed

(01-233072-011)

<u>Supervised By:</u>

Mr. Rizwan Aamir

A report is submitted to the department of Graduate Studies & Applies Science. Bahria University Islamabad campus. In partial fulfillment of requirement of the degree BS (CS)

Department of Graduate Studies & Applies Science

Bahria University Islamabad

ABSTRACT

This project is a design of MULTI_TOUCH SURFACE SYSTEM by FTIR (Frustrated Total Internal Reflection). An FTIR setup involves three vital components: a sheet of transparent acrylic, a chain of infrared LEDs, camera with an IR filter, and a projector. The LEDs are arranged around the outside of the sheet of acrylic so that they shine directly into the thin side surfaces. The foundation of the screen is the sheet of acrylic which serves as the medium for the infrared light. Acrylic has several properties that make it a good fit for our project. First, it has the right optical properties, allowing for an excellent FTIR effect. Additionally, it’s lightweight, strong, and very clear (more so than glass).

Once the IR light is inside the acrylic, it strikes the top and bottom surfaces of the acrylic at a near-parallel angle, and is subject to the effect known as total internal reflection. This causes it to be wholly maintained in the acrylic. The net effect of the setup described above is a sheet of acrylic full of internally reflecting infrared light. When a finger is pressed against the acrylic, it causes some of the light to be reflected down, through the acrylic and into the cabinet, where it is detected by the webcam. This effect, called frustrated total internal reflection is a little complicated, and involves something called an evanescent wave. The webcam, which has been modified to detect only infrared light. Software running on a nearby computer extracts a map of where fingers have been pressed on the screen, and uses that data to control multi-touch applications.

Finally there’s the cabinet—the structure that holds everything else together. The cabinet really only needs to do one thing: support the screen (including the acrylic, the lights, and the vellum surface) in such a location that's accessible to both the projector and camera. In an FTIR setup, the cabinet doesn’t even need to be closed.

For tracking and testing of the system the main software used is CCV (community Core Vision) [1]. Applications are developed in C# using Microsoft Visual Studio 2010.

ACKNOWLEDGEMENT

Thanks to our teachers for the encouragement and support provided by them especially to Sir Rizwan Aamir for giving us the idea and proving his all support that needed at every sharp edge of our project.

At this point special appreciation to the support and encouragement of Sir Aamir Iqbal Bhatti, Sir Adeel Nasir, Junaid Khalid and Zulqarnain Shehzad, thank them all for the humble support and specially the help provided by Sir Junaid Imtiaz throughout our journey till here, thank to them for their grace and also thankful to Bahria University for providing us the platform to express our self

LIST OF FIGURES

TABLE OF CONTENTS

Chapter #1

INTRODUCTION

1 INTRODUCTION

A new form of touch technology is a multi-touch computer display. It provides new ways to interact with computers, with the help of multiple touches along with many finger gestures. Multi-touch technology is becoming more and more admired due to the release of devices such as the iPhone™. Multi-touch display is made in a cost effective way by using Infrared (IR) light touch detection. The design uses low-cost materials and open source software to produce a multi-touch computer interface comparable to higher priced commercial products such as Microsoft® Surface™. This report will discuss the design of the multi-touch computer, encountered problems, design revision and software integration. It will also examine the commercial potential for the product and recommendation for the future designs.

Chapter # 2

Design & Overview

2 DESIGN OVERVIEW

There are certain ways to create a touch screen. The technology that is used in this project is FTIR (Frustrated Total Internal Reflection).In this project, it is IR (Infrared) light, that is reflecting internally within a part of acrylic, (Acrylic also known as Plexiglas).

IR-LEDs (Light Emitting Diodes) are positioned around the edges of acrylic, this way beam of IR light is made internally reflected inside the acrylic.

In an easy way, you can say that, IR light bounces within the acrylic sheet, from one end to another. As soon as your finger touches the surface of acrylic sheet, interruption of the internally reflected IR-light occurs. As show in figure:

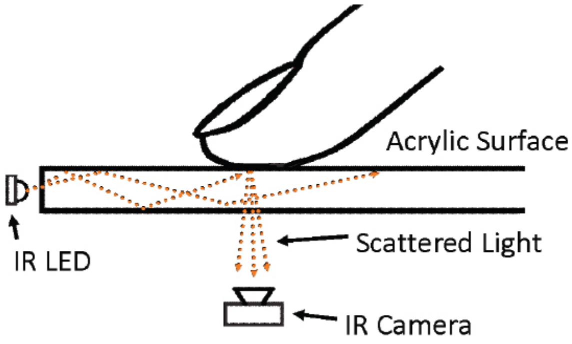


Figure 2.1 Frustrated Total Internal Reflection (FTIR) inside the Acrylic

The IR-light scatters on the finger tip. Infrared light is not visible to the naked human eye, but by situating an IR camera beneath the acrylic your, scattered IR light on the finger tip will be visible on the IR camera.

The scattered IR light on the finger tip is called as blobs. The image which is generated by the IR camera, include white blobs. These blobs are further analyzed by software.

Every blob created by the finger tip corresponds to certain coordinates. Software can perform certain tasks e.g. like moving, resizing or rotation of objects, by analyzing these coordinates

There's a projector located behind the acrylic screen, which is attached to a computer that projects the image on the screen.

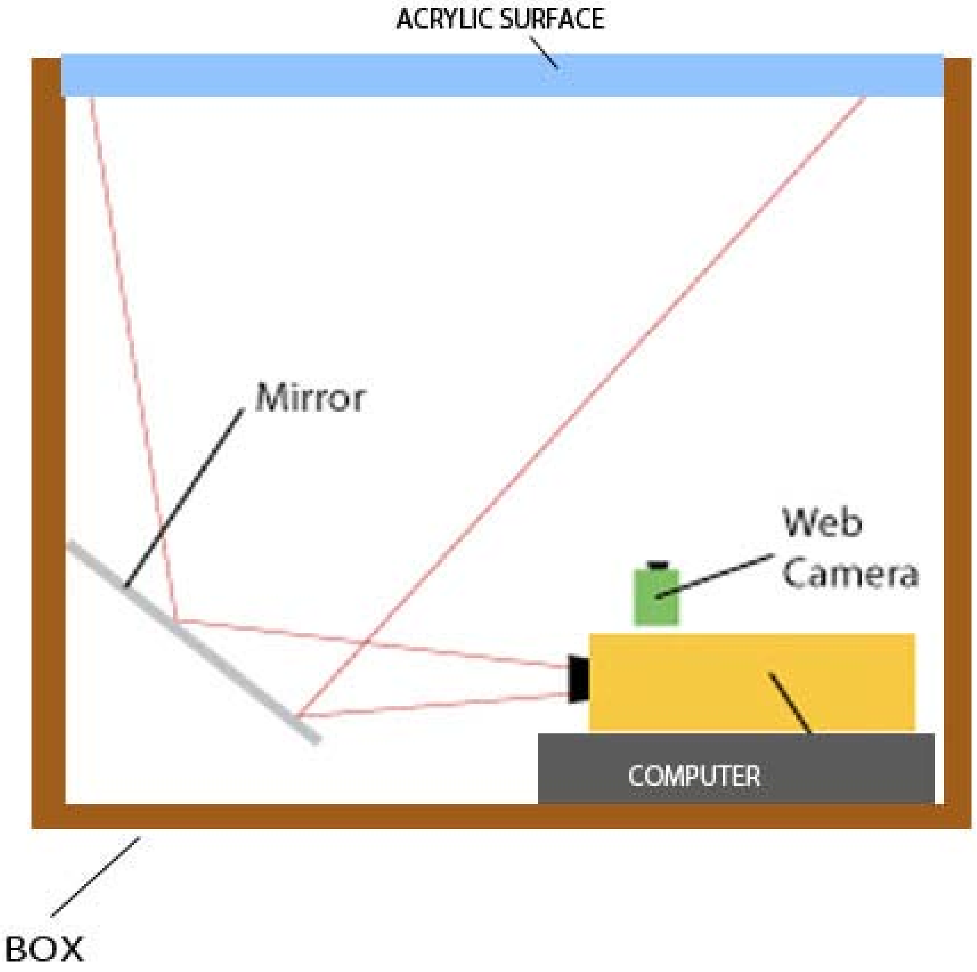


Figure 2.2 Overall Multi-Touch hardware designs

Chapter # 3

System Design

3 SYSTEM DESIGN

3.1 HARDWARE DESIGN

All the hardware involved in creating a multi-touch computer will be covered in this section. The hardware includes all the tools, materials used, cabinet, PS3 eye camera, acrylic, IR LED's, butter paper and the projector.

3.1.1 REQUIREMENTS

3.1.1.1 Tools

Standard size screwdriver (flathead & cross)

Small sharp knife

Scissors

Sanding paper

3.1.1.2 Building materials

Glue (hot glue)

Frame material (wood)

Electronic wire (0,5 mm)

IR-LEDs

Acrylic

Black photo negatives

Power source

Power cable

Playstation Eye USB camera

Projector

Projection material (tracing paper)

Working computer

Projector is provided to us by the university as it was the most expensive part of the device.

3.1.2 BUILDING PROCESS:

The hardware building process consists of the following 5 major elements

Table design and manufacture

Modifying a webcam

Treating acrylic

Wiring an electronic circuit

Projection

3.1.2.1 Table design and manufacture

The design of the table is built so that, the cabinet can supports the weight of a PC, a projector, acrylic screen and external pressure. The cabinet is made of 19x19 inch and ___ height.

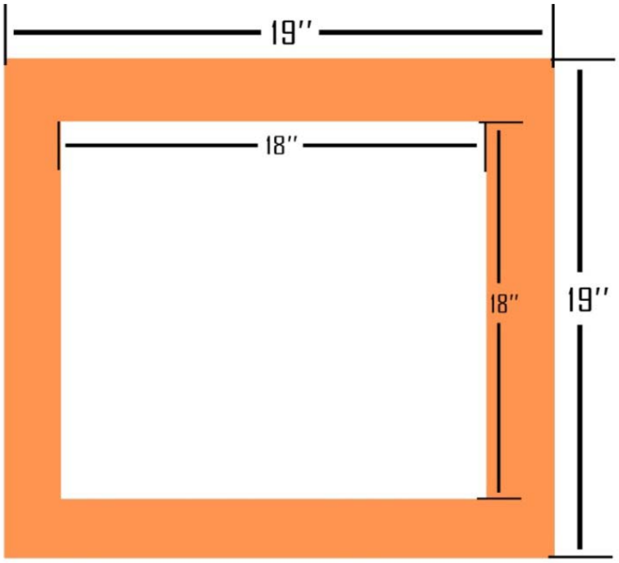


Figure 3.1 Cabinet Top view

3.1.2.2 Modifying the cam

Material needed:

· Webcam (used PS3 webcam, as it has enhanced frame rate then a usual webcam)

· Lens

· Magnetic Tape (From floppy disk)

· Tape, glue or any other stuff

Some webcam's detect IR, but it is not as clear as it is needed. Modification of a webcam is important to enable the detection of IR blobs that are generated by the finger tips.

3.1.2.2.1 PS3 eye cam surgery

In order to install the IR filter, PS3 Eye camera should be cracked first. To do this, first remove the four black disks glued over the screw wells.

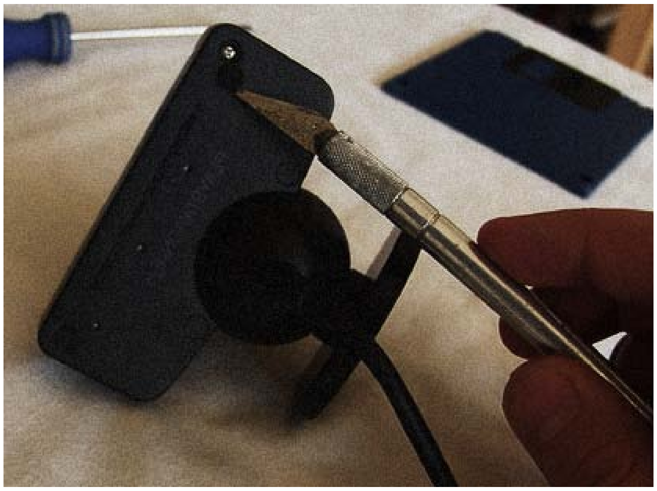


Figure 3.2 Opening PS3 Eye Cam

After the cover disk is gone screws can be easily removed, and after that open the camera from center using a flat screw driver. Circuit of the camera is exposed after the removal of back cover.

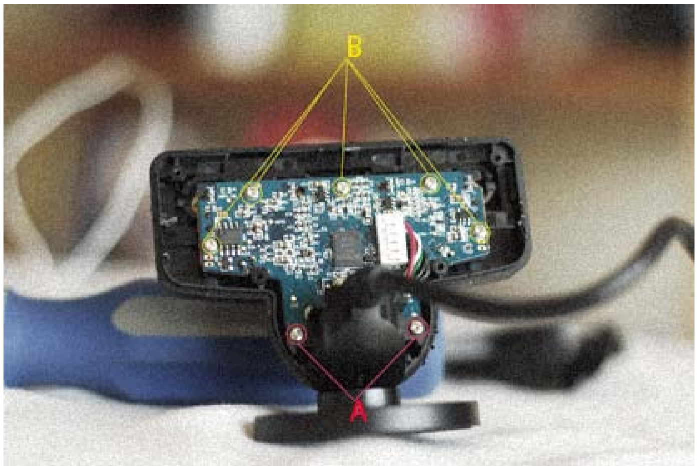


Figure 3.3 Inside PS3 Eye Cam

First remove screws labeled (a) in the figure which will remove the base. Then remove the screws labeled (b) which will separate the circuit from the casing. The IR sensor is at the innermost level of the glass.

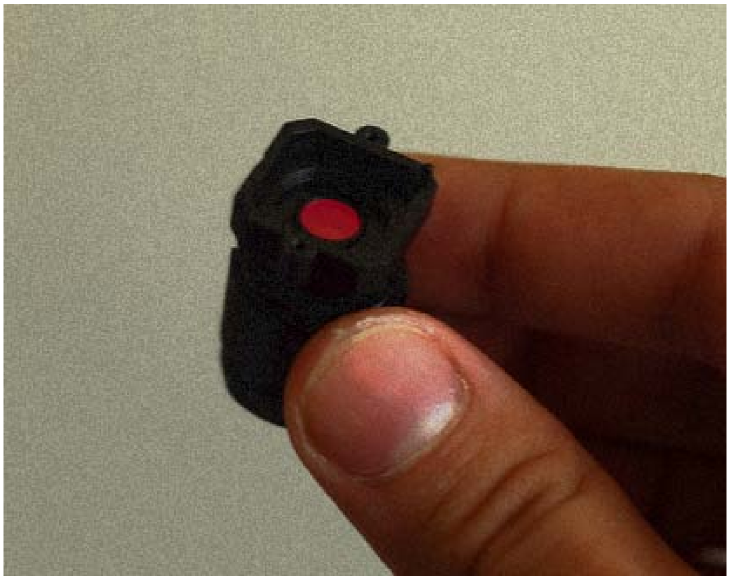


Figure 3.4 IR sensor inside PS3 Eye cam

The red thing is the IR sensor. Remove it and place the Magnetic disk cut out from the floppy disk of the right size to fit in.

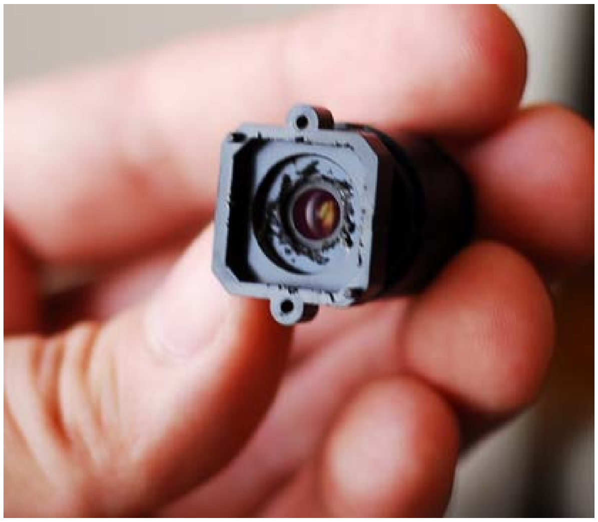


Figure 3.5 IR filter Removed

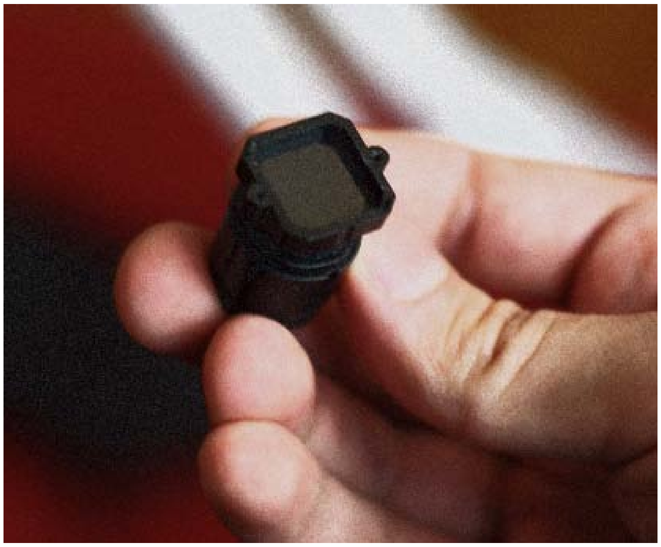


Figure 3.6 IR Filter Installed

PS3 Eye camera has two focus settings. It can be set by twisting the plastic ring around the lens. After placing the IR filter to detect the IR-light only, it messes up with its focus. Slightly twist the ring in order to get the specified focus for our purpose.

3.1.2.3 Treating acrylic

Acrylic is the material that is internally reflecting the IR-light. This material is very strong for this purpose. Acrylic is cut into the desired ratio that is 18x18’’ and 8mm thick. It should be at least 8mm thick, because LED’s that are placed of 8mm as well and it need to be robust, so if it is pressed even from middle it doesn’t break or bounce. When the acrylic is cut its edges become blind so, it needs buffing. Buffing the acrylic edges makes the edges transparent.

Figure 3.7 Polishing Acrylic

3.1.2.4 Wiring and placing the LED's

LED's needs a power source to light up. There are many ways to power up the LED's. For Example: by batteries (1.5 volt), USB (5 volt), 12 volt connecter on a computer, or a 12 volt adapter.

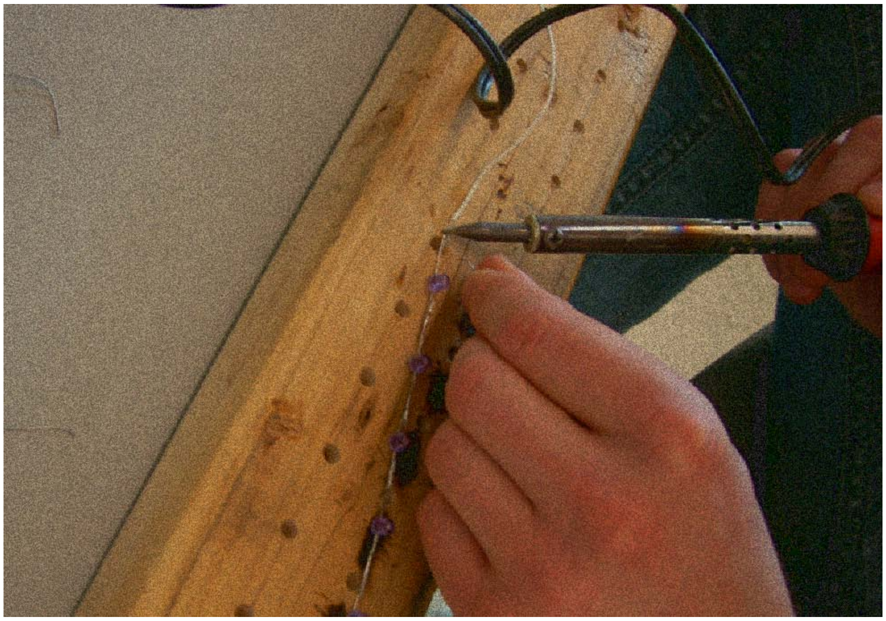


Figure 3.8 Wiring LED Circuit

In this project, 8 LED’s are placed in a row at all the four edges of the acrylic. Each strip is powered by a 12 volt 500 (mA) adapter.

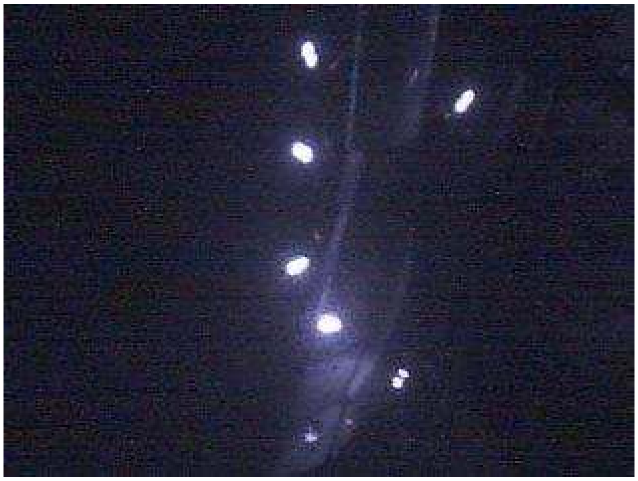


Figure 3.9 Working LED’s

Each led is soldered in a series circuit up to 8 LED’ in a row. LED’s are glued at the acrylic ends by hot glue, using glue gun.

After the connection of all the led’s is successful the testing of it is performed by turning on the cam beneath the acrylic and checking for fingers on the computer screen.

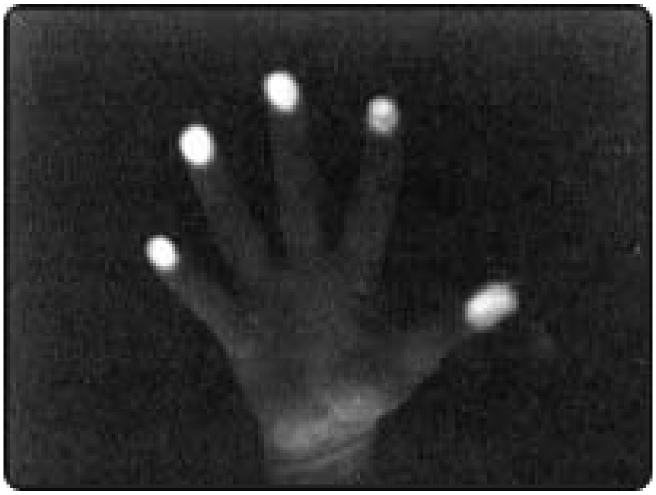


Figure 4.10 Hand Placed on screen, after FTIR functioning

3.1.2.5 Projection

A projector and a projection screen are used to project the image from a computer to the box. The projection screen is needed in order to display the projection from the projector, without it the projector beams will pass through the acrylic and no image will be shown. A projector screen is placed on top or underneath the acrylic. It should be placed on top of the acrylic, which has a plus point that you have direct interaction with the image.

In order to avoid a closed box solution, a mirror is used to reflect the image from the projector, just to shorten the distance.

Material used for projection screen is Tracing paper also known as butter paper. Tracing paper is cut out in the same ratio as of acrylic to fit on top of it.

The projector used is a normal 4:3 ratio projector.

3.2 DESIGN REVISIONS

3.2.1 Infrared Interference

One of the most concerning problem without multi-touch design was IR-light interference. When the hand is placed on the acrylic it detects some light and creates blobs before even the finger touches the surface. This problem was found at the corners of the acrylic. After testing and troubleshooting the problem detected was that IR light emitting through the LEDs were reflecting on the hand directly. Light was bouncing inside the acrylic rest of the light from led’s which was emitting was reflecting on the hands and being detected by the camera as well which made the detection malfunction.

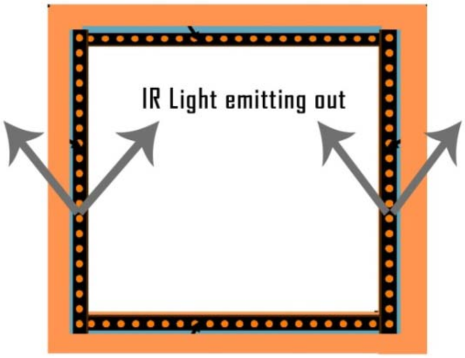


Figure 5.11 IR light interference

It was treated by covering the LEDs by placing hardboard material on top of the led’s at all the four corners. So the light which was emitting was stopped by blocking it.

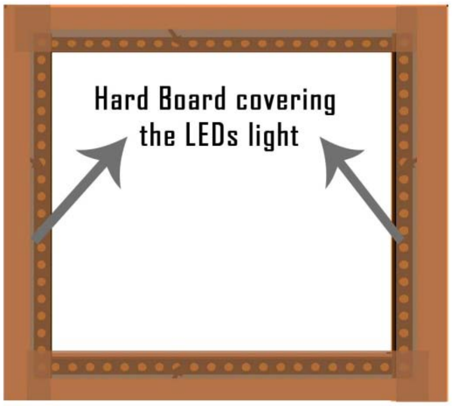


Figure 3.12 IR light interference resolved.

3.2.2 Overheating inside the cabinet

The projector placed inside the cabinet gives a large amount of heat and it is not designed to be placed inside a closed box of this size. Most of the projectors have heating issues and turn off when it heats up. When the testing was going the projector turned off several times and it was of a big problem. To get rid of this issue a small computer fan is placed in the cabinet. A hole was cut out of the box near the projector to provide exhaust for air to pass through.

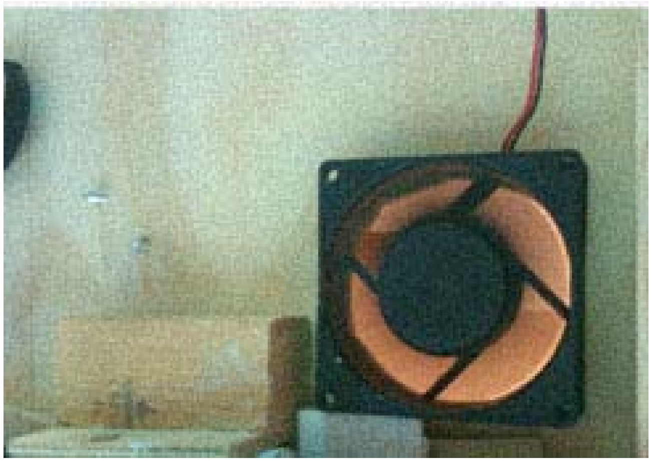


Figure 6.13 Fan placed inside the cabinet

3.3 SOFTWARES DESIGN

3.3.1 Introduction

To communicate between a multi-touch display and a computer, computer applications are necessary.

The main software that is used in this project is Touchlib, it is an open source library which takes the visual data from the camera and parse it into touch events, which is further used by different applications and programming languages to provide multi-touch control. This library directly, allows implementation of standalone multi-touch apps.

Application development is done in C # using Visual Studio 2010.

3.3.2 CL-EYE Driver (Code laboratories)

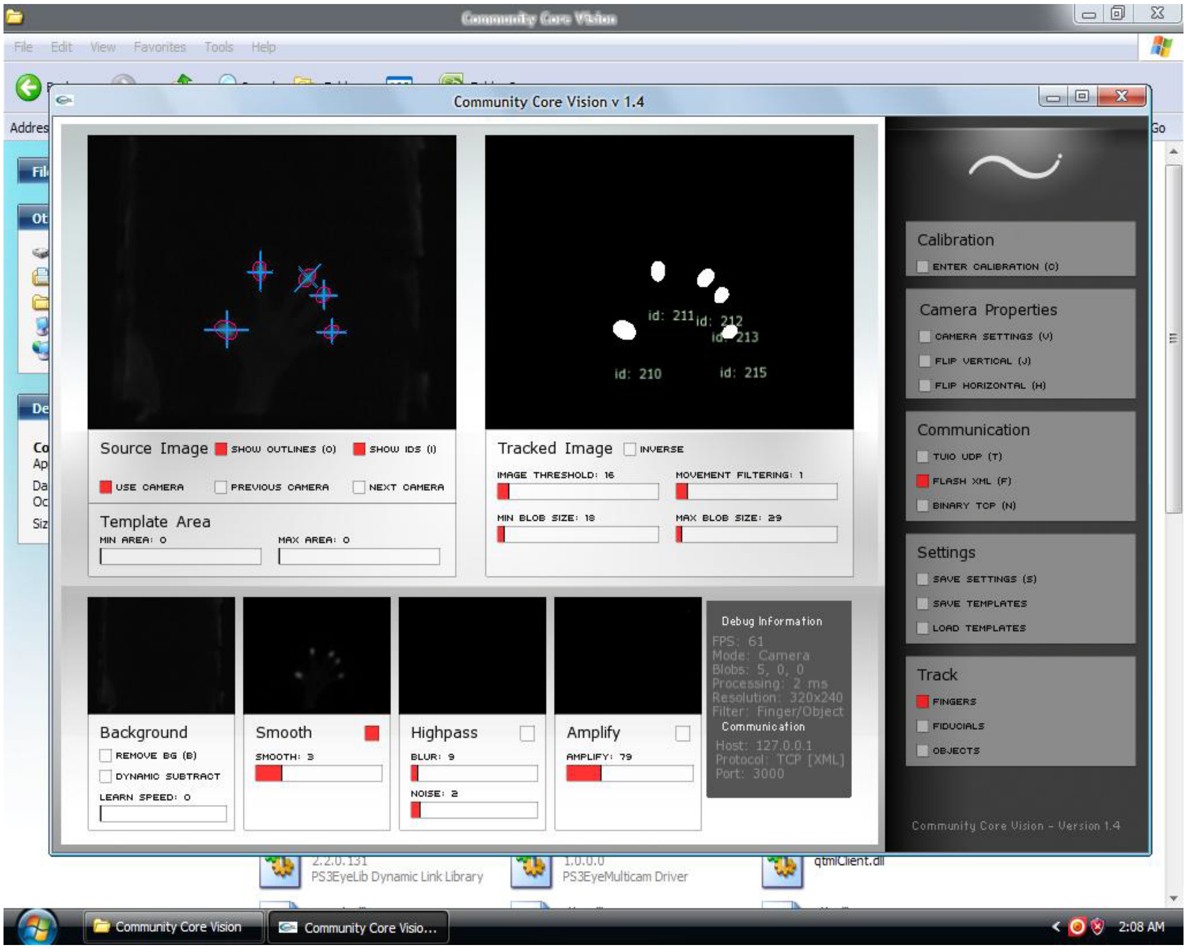
Sony Playstation3 camera (Playstation eye) was not intended to use with any other thing other than that of Playstation 3. So drivers are needed to make it in use.

Alex Popovich[2] developed CL-eye driver in beginning of 2008, with build (2.0b81016) which is working fine with Windows and Community Core Vision (CCV) library released on October 17, 2008. “As advertised” this driver operates with minimal latency (1-10ms on PC)

Camera should be connected to USB controller with no other USB peripherals connected to that controller in order to provide the camera with maximum bandwidth.

3.3.3 Community Core Vision (CCV)

After the camera is set up to sent the raw image data into the computer software is needed to transform that image data into input data. This is handled by CCV open source library to handle touch events developed by Natural User Interface group [1] (NUI). An image of the interface is shown in fig 3.10. Touch lib provides configuration and operation of the camera with config.xml file which set up to 640x480 @ 60 fps.



Source Image 🟥 SHOW OUTLINES (O) 🟥 SHOW IDS (I)
🟥 USE CAMERA □ PREVIOUS CAMERA □ NEXT CAMERA

Template Area
MIN AREA: 0 MAX AREA: 0

Tracked Image □ INVERSE
IMAGE THRESHOLD: 16 MOVEMENT FILTERING: 1
MIN BLOB SIZE: 18 MAX BLOB SIZE: 29

Background
□ REMOVE BG (B)
□ DYNAMIC SUBTRACT
LEARN SPEED: 0

Smooth 🟥
SMOOTH: 3

Highpass □
BLUR: 9
NOISE: 2

Amplify □
AMPLIFY: 79

Debug Information
FPS: 61
Mode: Camera
Blobs: 5, 0, 0
Processing: 2 ms
Resolution: 320x240
Filter: Finger/Object
Communication
Host: 127.0.0.1
Protocol: TCP [XML]
Port: 3000

Calibration
□ ENTER CALIBRATION (C)

Camera Properties
□ CAMERA SETTINGS (V)
□ FLIP VERTICAL (J)
□ FLIP HORIZONTAL (H)

Communication
□ TUIO UDP (T)
🟥 FLASH XML (F)
□ BINARY TCP (N)

Settings
□ SAVE SETTINGS (S)
□ SAVE TEMPLATES
□ LOAD TEMPLATES

Track
🟥 FINGERS
□ FIDUCIALS
□ OBJECTS

Figure 3.14 Community Core Vision Interface

CCV accomplishes two purposes. It adds filters to image data received from the camera. First it applies important filter to camera to convert RGB data into black and white image to make the touched and untouched surface easily distinguished. Different filters added are shown in the previous figure (fig 3.10). With our setup on, following settings are to be chosen:-

3.3.3.1 Configuration Process

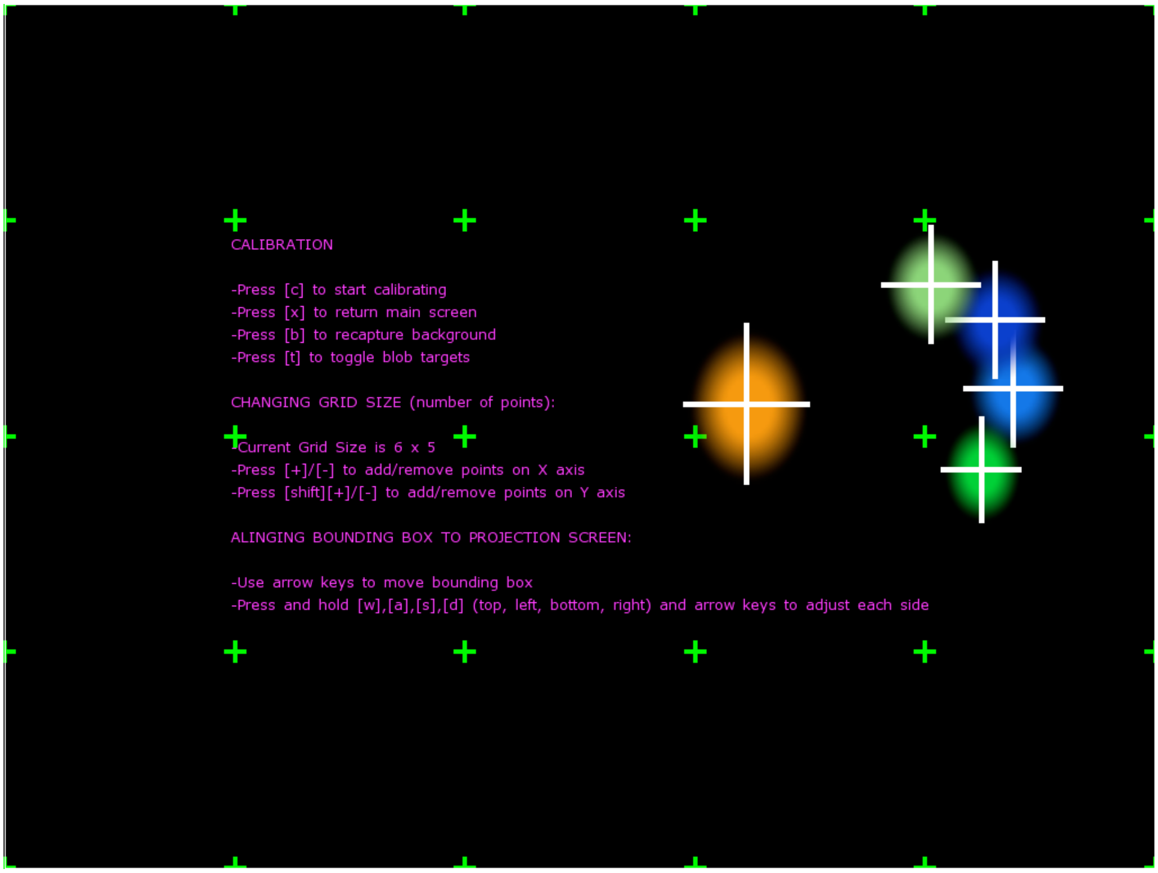
- Background Functions
  - Remove Background function provides the software a reference IR lighting pattern to compare the input with. It removes any unwanted background IR

blobs, while nothing is touching the surface. Pressing B at anytime during configuration capture the background.

  - Dynamic Subtract Function: This functionality was ideal for our setup because of the environment and the light in which it is placed in. Whenever the image on the screen changes the brightness and when the external light changes it causes false blobs to appear. Dynamic subtract function has the ability to balance quick removal of fake blobs and has adjustable learning speed while tracking the touch input accurately.

- Smooth, Highpass and Amplify Filters:
  - Smooth Filter: smoothes and removes noise from the image. Adjust it until there are no more white dots on the screen.
  - Highpass Filter: It is turned off, as the setup was design for FTIR not diffused IR.
  - Amplify Filter: amplifies the blob, adjustment of it is done if the blobs are very weak or requires a lot of pressure to create a blob. Turned it off as the blobs created were fine enough and amplifying it was allowing the interference of fake blobs.
- Tracked Image Functions:
  - Image Threshold: adjust the levels of image threshold
  - Movement Filtering: adjust the detection and threshold of moving touches.
  - Min/Max Blob Size: specifies the size of the blobs to detect by the software. Adjust it so the number of fake blobs is minimized.
- Calibration
  - After CCV is tuned and set up to detect best IR touches without background distortion, calibration is required in order to relate the capture data with the blob data from the screen. CCV provides calibration utility residing within the configuration process.
  - Press C when all the settings has been set, no fake blobs are shown in the window and the touches are visible.

- Calibration utility is very useful as it not only relate touch points, but can also transform the images from the camera that might be flipped relative to the screens output. This is a very compulsory step as it is free to locate the camera where it wants. Calibration screen can be seen in the figure 3.11
- Press C to start the calibration. There are several points given in the calibration screen (represented with + symbol).
- When the calibration starts each + point is highlighted one by one. That point should be pressed for a few seconds to obtain accurate capture. Capturing the accurate points according to your screen is very necessary.
- When all the points have been captured press Touching the surface will create colored blobs as shown in the figure below
- Press ESC to update the config.xml file



```
CALIBRATION

-Press [c] to start calibrating
-Press [x] to return main screen
-Press [b] to recapture background
-Press [t] to toggle blob targets

CHANGING GRID SIZE (number of points):

-Current Grid Size is 6 x 5
-Press [+]/[-] to add/remove points on X axis
-Press [shift][+]/[-] to add/remove points on Y axis

ALINGING BOUNDING BOX TO PROJECTION SCREEN:

-Use arrow keys to move bounding box
-Press and hold [w],[a],[s],[d] (top, left, bottom, right) and arrow keys to adjust each side
```

Figure 3.15 Calibration Screen

3.3.4 Table-Top Tangible User Interface (TUIO)

TUIO[3] protocol is designed for table top user interfaces and to meet their requirements by presenting communication between an application layer and a controller interface. It was originally written by Martin Kaltenbrunner and Ross Bencina of University of pompeu Fabra in Barcelona, Till Bovermann of Bielefeld University in Germany, and Enrico Coastanza of Medialab Europe in Dublin Ireland[4] presented in the paper in 2005. The latest version of this software available is 1.1. It uses Open Sound Protocol (OSC) [5]. Since a wide range of platforms already supported OSC, TUIO was made efficient to be used by any of the platforms. At the time when paper was first written, the authors were working for Java, PureData, Max/MSP, Flash and C++ implementations. In order to provide low latency communications which is necessary for real-time data source (in our case camera) TUIO uses the UDP communication protocol.

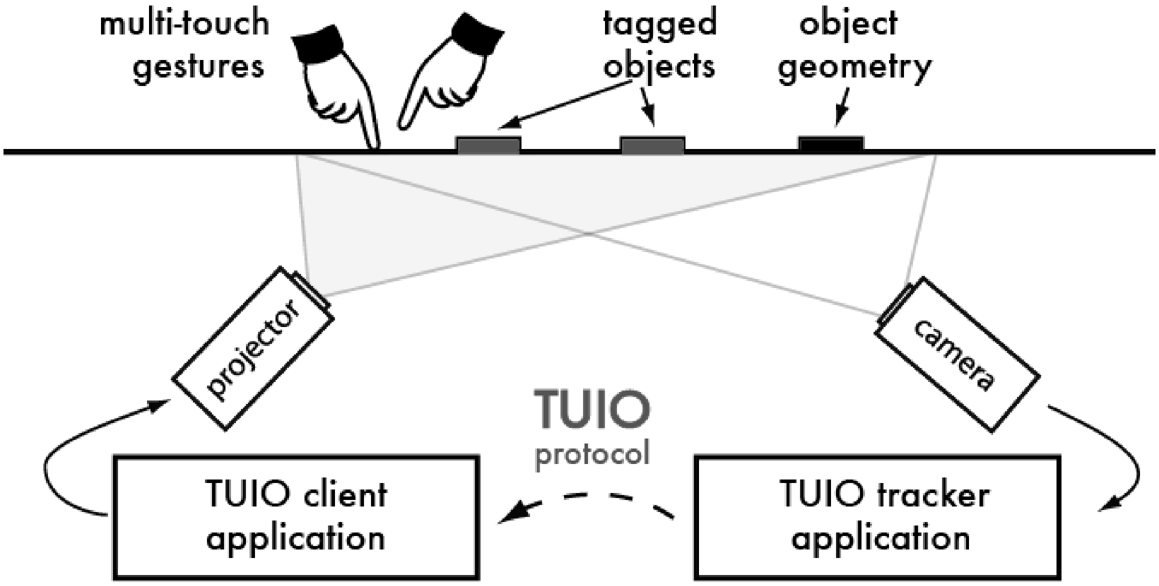


Figure 3.16 TUIO Working

For TUIO tracker application CCV (community core vision) is used and for the TUIO client application C# is used to build TUIO listener applications.

3.3.5 Adobe Flash

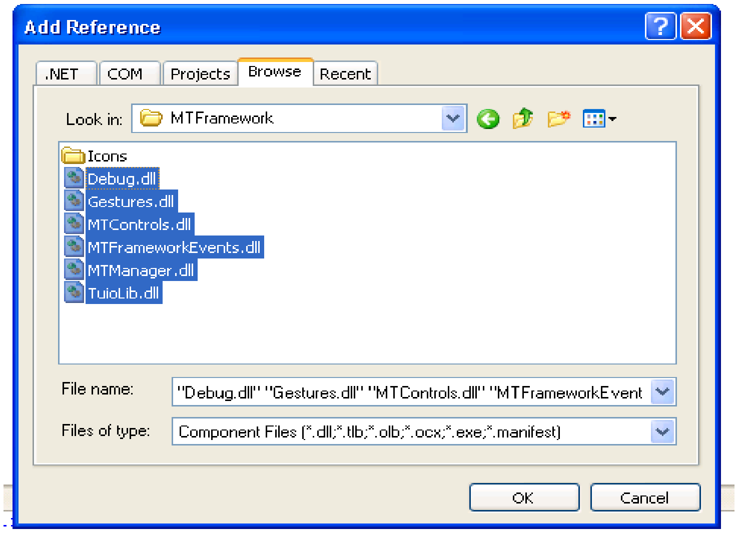
For creating demonstration application for multi-touch interfaces Adobe Flash is an excellent tool. It is relatively very easy to develop graphically oriented applications in it using Adobe Action script, and is an OSC and TUIO supported platform.

CCV provides a couple of flash demos which were executed and tested. These demos are very useful for testing of the FTIR screen.

3.4 WPF MT FRAMEWORK

The Multitouch (MT) framework needs a server to work properly because it is based on TUIO (Tangible User Interface) protocol. As mentioned above, in this project CCV (community core vision) for the server needed. This framework work only with the Microsoft .net framework 3.5, Microsoft Expression Blend and Microsoft Visual Studio 2008 and above.

Six DLL’s needed to be added to the project in order to use the MT Framework



Look in: MTFramework

Icons
Debug.dll
Gestures.dll
MTControls.dll
MTFrameworkEvents.dll
MTManager.dll
TuioLib.dll

File name: "Debug.dll" "Gestures.dll" "MTControls.dll" "MTFrameworkEvent

Files of type: Component Files (*.dll;*.tlb;*.olb;*.ocx;*.exe;*.manifest)

Figure 3.17 MT Framework

3.4.1 Getting Started

To make an MT application, there is some code to be added first. But default grid should be removed which is automatically added as it is not MT Enabled so it will not work for multi-touch. Vital thing is that a MT Application ALWAYS need a MT-Grid or a MT-Canvas as parent and it needs to be of the same size, else there will be malfunctioning.

Make a MT-Grid or a MT-Canvas and give it a name. After that xaml code might look like this.

```
<Window x:Class="MultitouchApp.Window1"
   xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
   xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
   Title="Window1" Height="300" Width="300" xmlns:my="clr-
   namespace:MTControls;assembly=MTControls">
   <my:MTCanvas Name="MTCanvas"></my:MTCanvas>
</Window>
```

MT-Canvas has to be linked to the class for what is receiving the points from the server which is MT-Processing. This class does some moving, rotating and other calculations on objects by calculation what and where an object has been touched by a finger.

There are some dependency properties that can be set for each object. These properties are: “CanBeClicked”, “CanBeScaled”, “CanBeDragged” and “CanBeRotated”.

3.4.2 Supported Controls

MT control functions are

- **CanBeClicked** (dependency property)

  When this property is enabled, a click event will be triggered at the object. And when it is disabled, there will be no click event generated.

- **CanBeDragged** (dependency property)

  In MTCanvas user has the possibility to move and drag object all over the window

  In MTGrid case user can drag and move object from one column or row to another. If there is more the one rows or columns.

- **CanBeScaled** (dependency property)

  MTCanvas: in MTCavas it is possible for user to scale an object whereas in MTGrid: it is not possible to scale an object.

- **CanBeRotated** (dependency property)

  MTCanvas: user has the flexibility to rotate an object when it is placed in MTCanvas.

  MTGrid: rotation of an object is not possible in MTGrid.

- **Timeout (int)**

  When an object is touched by finger a timer starts and triggers when timeout has been arrived. NotMoved events uses this property.

- **AdornersEnabled (bool)**

  Enabling this property will make some detailed information visible about the object when it is touched. This info consists and varies on the dependency properties like CanBeMoved, CanBeDragged etc

- **Not Supported Properties**

  Some of the Non Supported dependency properties till yet are:

  - IsScaling
  - IsRotating

- IsDragging

3.4.3 Gestures Module

Gestures Module by default recognizes some gestures. It can be enabled or disabled by changing gesture enabled Boolean. The default gestures are

3.4.3.1 Scale

In order to perform a scale action at least a touch from two fingers is required. It is done by dragging out two fingers from each other. Another form of scale is dragging away which is done with one finger

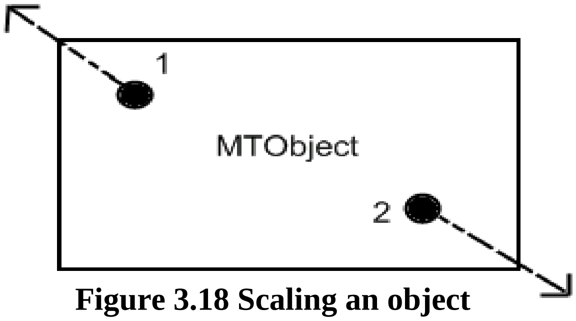


Figure 3.18 Scaling an object

3.4.3.2 Rotate

Minimum of two finger touch is required in order to perform a rotate operation. Rotate is not a real gesture but a move of two finger is.

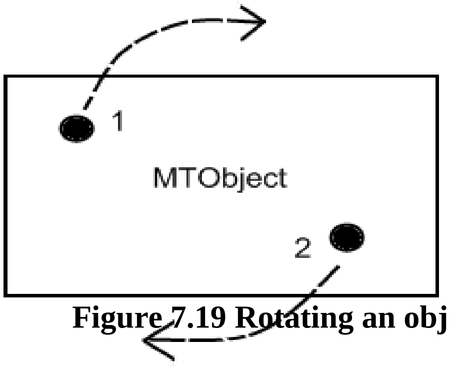


Figure 7.19 Rotating an object

These Gestures will be recognized by gesture library

Lightning

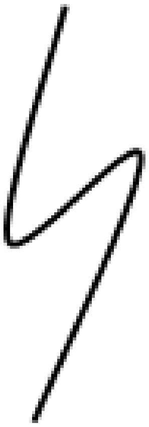


Figure 3.20 Lightning

CircleCCW

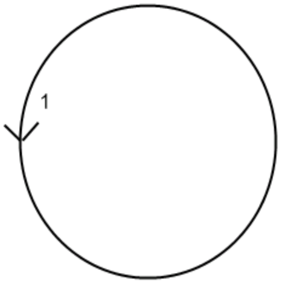


Figure 8.21 Circle CCW

CircleCW

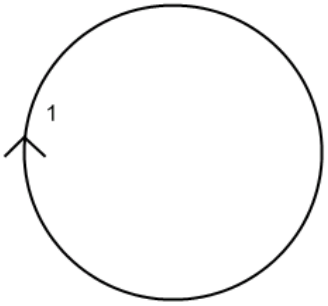


Figure 3.22 Circle CW

DualCircleCW

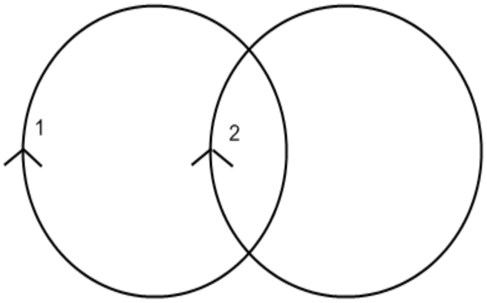


Figure 3.23 Dual Circle Cw

DualCircleCCW

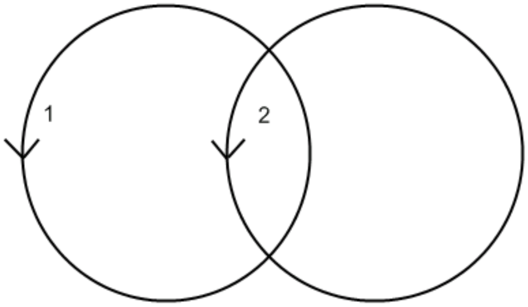


Figure 3.24 Dual Circle

TriangleCW

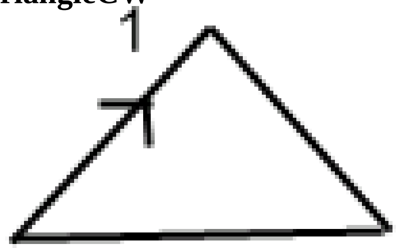


Figure 3.25 Triangle CW

TriangleCCW

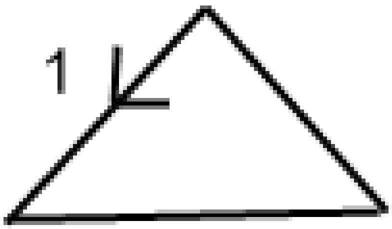


Figure 3.26 Triangle CCW

DualTriangleCW

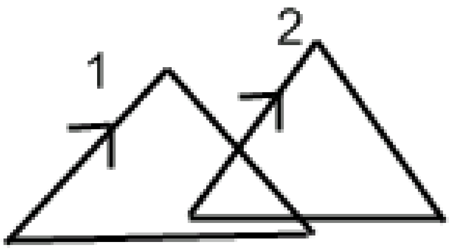


Figure 9.27 Dual Triangles CCW

DualTriangleCCW

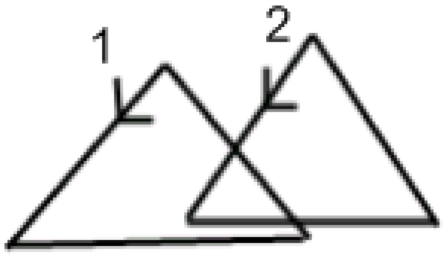


Figure 3.28 Dual Triangles CCW

3.4.4 Known issues and bugs

Disconnect listening thread (application doesn't close)

Disconnection of the listener thread has to be done manually. Disconnect method in MTProceesing will help in this matter.

Click event

A click event will be triggered two times when simulating with the mouse. E.g. button click will be raised double when clicking with mouse on a button in simulation mode. At Multi-touch Table it works fine.

Combobox

The problem with Combobox is this that it does not expand too well.

3.5 PROCESS MODEL

Every project needs a process model to be followed in order to acquire an accurate result at the end. So, in this project WATERFALL PROCESS MODEL has been followed. Waterfall process model works sequentially, often used in software development process, in which progress is curving gradually downwards. Waterfall process model has seven phases which should be followed in software development.

3.5.1 Phases of Waterfall Process Model

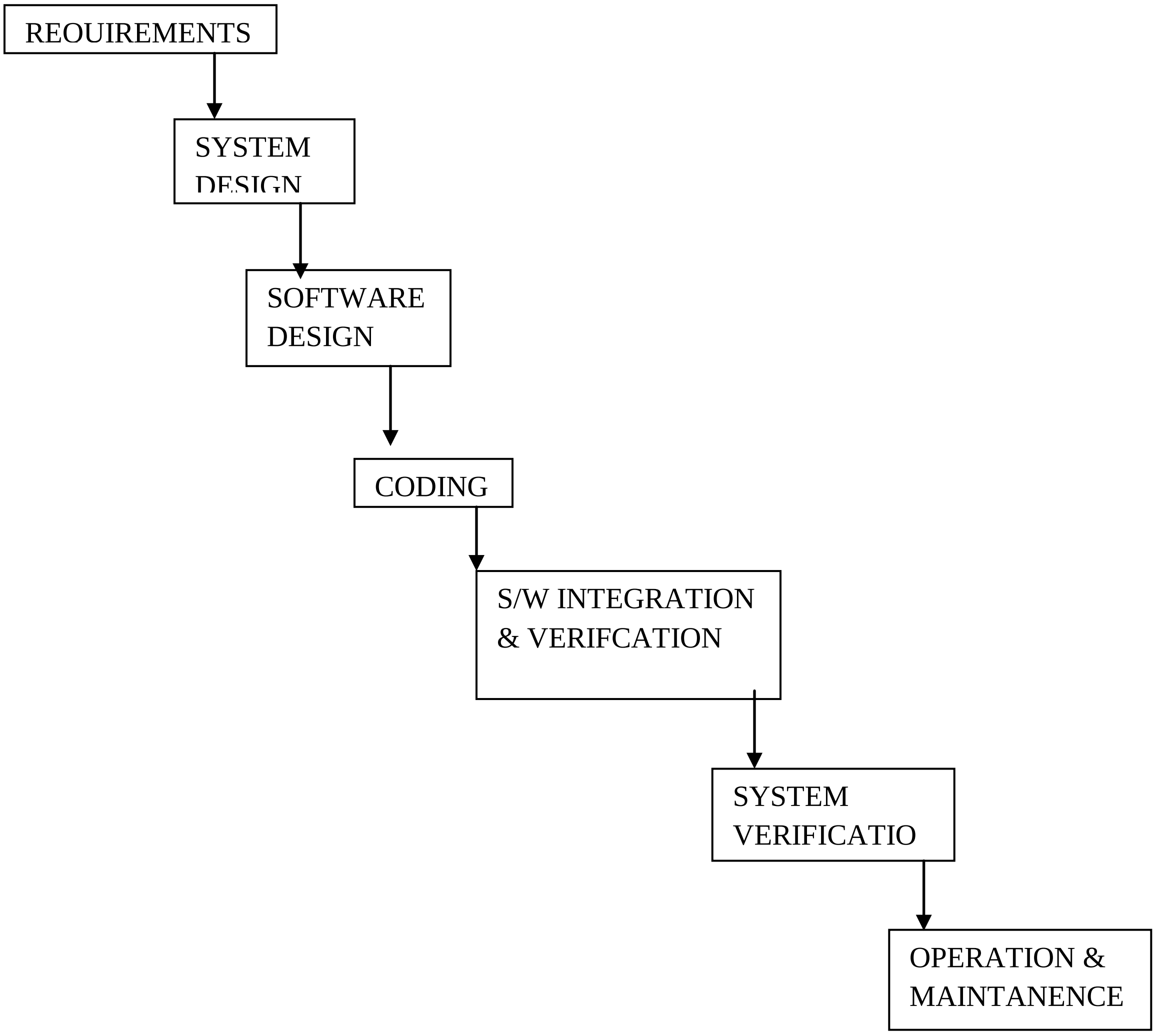


Figure 10.29 Waterfall Model

3.5.1.1 Requirements

All the necessary requirements of the system which is going to be designed are gathered in this step. Requirements have to be gathered by evaluating the needs of the end user and checking them for validity and possibility to implement them. The mission is to make a Requirements Specification Document which is used to the input of the next phase of the model.

3.5.1.2 System Design

The structure has to be accurately designed before any implementation is started. This includes an architectural design which describes the main components of the structure, their interfaces and interactions. By this hardware is defined and software is divided in its components. E.g. this includes the selection of computer platform, an operating system, other peripheral hardware, etc. The software components have to meet up the user's requirements and to meet the need of possible scalability of the system. Generation of the System Architecture Document is the aim of this phase which is an input for the software design phase of the development and also for hardware design.

3.5.1.3 Software Design

The software design will break the main software blocks into code modules which is based on system architecture. In this phase interfaces and interactions of the modules are described and their functional contents as well. Startup, shutdown, error conditions and diagnostic modes and all the necessary system states has to define. Software Design document is an output of this phase.

3.5.1.4 Coding

In this phase, setting up the defined modules based on software design document and actual coding is started. First the system is developed in smaller portion called units. They are integrated later on if they are able to stand alone from a functional aspect.

3.5.1.5 Software Integration & Verification

Unit is independently developed and tested for its functionality. This testing is called unit testing. This test verifies that units are meeting their specifications. In integration process developed and tested units are brought together then it checks the whole module as it is working or not.

3.5.1.6 System Verification

After successfully integration complete system should be tested against its initial requirements. This will include the original hardware. But the previous integration and testing may still be performed.

3.5.1.7 Operation & Maintenance

Operation and Maintenance is a never ending phase. All problems will be solved in this last phase. In this phase system is handed over to the customer and customer checks its requirements. If requirements do not meet the customer’s need then it has to be fixed to make the system usable.

Chapter # 4

Testing

4 TESTING

4.1 Phase 1

- When acrylic is bought its one or more than one side is blind (i-e it cannot pass the light into the surface). It is then polished using sanding paper. It is actually a really hard task. After all the sides of the acrylic are made polished, then it is being tested whether it is passing the light into the surface or not. This is done by passing any visible light, mostly from torch on the sides of the acrylic in a dark room. When the light is passed into the surface from one end, it can be clearly seen leaving the surface from the other side.
- After the filter is installed in PS3 Eye camera, testing of it is needed to check whether it is detecting the IR light or not. Testing of the camera can be done by simply turning on the camera and checking by clicking on a working TV remote control facing the camera.
- Once the LEDs are connected in series. 8 LEDs in each side of the acrylic. It is been tested by the PS3 Eye cam which is now modified to detected only IR light.

4.2 Phase 2

- Once FTIR screen is setup and the camera is placed under the surface to detected blobs. IR leaking the surface can be seen from the bottom. The IR interference which is already discussed in 3.2.1 Infrared Interference is tested by covering one by one each side of the L.E.D's and then again checking for the leaking by moving hand on top of the surface and touching it too.
- Testing of the box is also done by moving the camera all along side for checking whether all the L.E.D's are working or not. And the light is leaking or not.

Chapter # 5

Future Works

5 FUTURE WORKS

Designing a Multi touch surface system is a very vast kind of project. Adding more and advance applications will make it more striking. In this project few applications were added so limited work could be done by this project. Following features could be added in future:-

Air Movement Detection: This technology detects when someone blows onto the surface, it measures the intensity and direction of the air flow.

3D Hand Tracker: It provides touch free cursor control (which is based on windows applications) by measuring the position of user’s hand(s). One camera or detector would be placed outside the table which detects the human hand(s) within an adjustable volume and distance from the camera.

Future works aims to improve the robustness of the project.

Chapter # 6

Commercial Potential

6 COMMERCIAL POTENTIAL

Currently there are not much competitors in the business as Multi-touch computing is comparatively new field. Users can interact with each other and the computer on a large table surface using our display. Most of the touch screens available today offers one touch control. Many of the companies still function with keyboard and mouse. Commercially available multi-touch screens in the industry now a days are Microsoft Surface, the Mitsubishi Diamond Touch and Reactable.

The Technology used by Microsoft Surface includes HQ projector, 5 IR cameras to recognize object on the screen. It costs $12,500.

The Mitsubishi Diamond Touch provides touch sensing through capacitive coupling. The Diamond Touch costs $9,500 but does not include the cost of the projector or computer.

The Reactable Surface consists of an acrylic sheet on top and short-throw projector. Tracking is done through software capabilities and tagging. The future price for Reactable is unknown and is currently not for sale.

Chapter # 7

Recommendations

7 RECOMMENDATIONS

The main concerning problem in IR multi-touch design is ambient IR interference. The ambient interference was an issue in new environments. Some IR filtering out was integrated in our project but still there were some problems regarding ambient IR interference. A screen consists of projection surface and IR filter was produced by Rosco called Rosco Grey [6], it was not tested in our project, so it may or may not filter out all IR light.

Chapter # 8

Conclusion

8 CONCLUSION

The main objective of this project was to make a multi-touch system that is much lesser in cost than other multi-touch systems in the current market. This project was built using inexpensive materials and open source software, further applications were developed using C# in Microsoft visual studio 2010. The main concerning issue was the IR ambient light which affected the project with the changing environment. Some actions were taken to minimize the interference of IR light from external sources but further work can be done to eliminate IR interference in changing environment.

REFRENCES

1. Community Core Vision and NUI Group homepages: http://nuigroup.org

http://ccv.nuigroup.org

2. Code Laboratories Home Page, Alex Popovich: http://codelaboratories.com

3. TUIO 1.1 Protocol Specification: http://www.tuio.org/?tuio11

4. TUIO: A Protocol for Table-Top Tangible User Interfaces, 2005: http://mtg.upf.edu/files/publications/07a830-GW2005-KaltenBoverBencinaConstanza.pdf

5. Features and Future of Open Sound Control version 1.1 for NIME, Adrian Freed and Andy Schmeder, USC Berkley, 2009 http://cnmat.berkeley.edu/system/files/attachments/Nime09OSCfinal.pdf

6. Rosco Grey Pojection Screen: http://www.rosco.com/us/screens/roscoscreen.cfm#Grey