

## List of Figures

1	Laser Pointer based Interactive Projector System . . . . .	1
2	Hardware and Electrical parts of Looma system . . . . .	3
3	3D printed hacked Wii Remote wand . . . . .	4
4	Illustration of how the movement of infrared is picked up by the Wiimote . . . . .	5
5	Block Diagram of the working of our system . . . . .	6
6	Looma Software . . . . .	6
7	Proposed Block Diagram . . . . .	8
8	Intensity Gradient and Direction . . . . .	11
9	Histogram Analysis of Laser Pointer . . . . .	12
10	555 Timer Calculation . . . . .	13
11	Circuit Test on breadboard . . . . .	13
12	Distorted Projected Screen . . . . .	14
13	Perspective Transform Correction . . . . .	15
14	Laser circuit schematic diagram . . . . .	15
15	Laser circuit eagle design diagram . . . . .	16
16	Laser circuit front . . . . .	16
17	Laser circuit back . . . . .	16
18	State Transition Diagram of Algorithm . . . . .	17
19	Works Completed Timeline 7th Semester . . . . .	20
20	Works Completed Timeline 8th Semester . . . . .	20
21	Works Remaining Timeline 8th Semester . . . . .	21

# Contents

<b>List of Figures</b>	<b>i</b>
<b>Contents</b>	<b>ii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Background . . . . .	1
<b>2 Objectives:</b>	<b>2</b>
<b>3 Literature Review</b>	<b>3</b>
3.1 Looma and Wand Details . . . . .	3
3.2 System Operation Description . . . . .	4
3.3 Problem Statement . . . . .	4
<b>4 System Block Diagram</b>	<b>6</b>
4.1 System Block Explanation . . . . .	6
4.1.1 Projected Screen of Looma . . . . .	6
4.1.2 Raspi Camera Module . . . . .	7
4.1.3 Using Picamera . . . . .	7
4.1.4 Raspberry Pi . . . . .	7
4.2 System Design and Description . . . . .	7
<b>5 Works Accomplished in the Seventh Semester</b>	<b>10</b>
5.1 Detection of Laser Pointer and Mouse Movements . . . . .	10
5.2 Dynamic Exposure Correction . . . . .	12
5.3 Porting of Code to Raspberry Pi . . . . .	12
5.4 Design Calculations of 555 timer . . . . .	12
<b>6 Works Accomplished in the Eighth Semester</b>	<b>14</b>
6.1 Calibration of Projected Screen . . . . .	14
6.2 Finalization of Laser Detection Algorithm . . . . .	15
6.3 Microcontroller to generate laser pulses . . . . .	15
6.4 Finalized algorithm for click and drag . . . . .	16
6.5 Communication between computer and RPi . . . . .	17
6.6 Documentation . . . . .	18
<b>7 Works Remaining in the Eighth Semester</b>	<b>19</b>
7.1 Effective Algorithm implementation . . . . .	19
7.2 Using the mouse library . . . . .	19

7.3	Integration of the system . . . . .	19
7.4	Documentation . . . . .	19
<b>8</b>	<b>Project Timeline</b>	<b>20</b>
8.1	Works Completed Timeline 7th Semester . . . . .	20
8.2	Works Completed Timeline 8th Semester . . . . .	20
8.3	Works Remaining Timeline 8th Semester . . . . .	21
<b>9</b>	<b>Bibliography</b>	<b>22</b>
<b>10</b>	<b>References</b>	<b>23</b>

# 1 Introduction

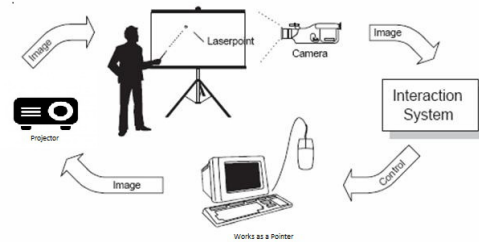
## 1.1 Background

Projector systems project on the screen the contents of the computer connected to it. By the use of various technologies, we can increase the interactivity of projector systems.

Looma is a portable projection system that uses an infra-red wand to navigate the screen. This system was built by Village Tech Solutions(VTS). Looma is being researched to make it affordable and low power consuming audio-visual technology device which will provide an interactive window to internet and access to educational contents to village schools that have never seen computers, or in some cases, even books.

Our project is trying to improve the interactivity and the cost of the existing Looma system such that it can be afforded by rural schools. If there is a camera used to capture the projection area, and which locates the position of the pen or other visual tools, this would be an interactive projector system. With the use of a camera, and laser pen, our project gets scoped under **Human Computer Interaction(HCI)**. **Computer Vision(CV)** is an integral part of the interaction.

This vision based HCI acts as an apt mediator between the user and the computer. Here, along with images and their enhancements, the system would be emulating human vision(examples: object recognition, tracking,etc.).



Laser Pointer based Interactive Projector System

## 2 Objectives:

The basic objectives of our project is the modification of Project Looma in aspects listed below:

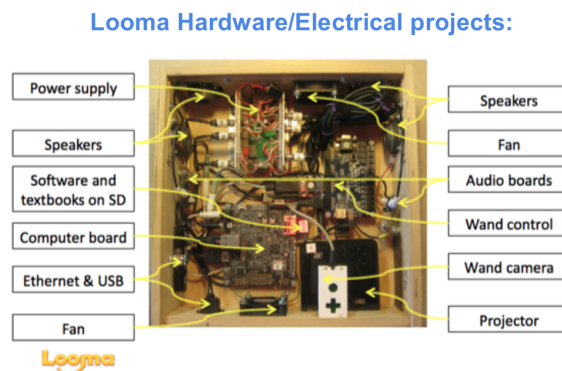
- To make prototype for an alternative type of vision-based HCIs (Human Computer Interaction)
- To use laser pointer instead of IR (Infra-red) based interactive tool
- To remove the Wii technology, and its dependencies used in current prototype thereby, making the system efficient and cost effective

### 3 Literature Review

The current prototype can run off a rechargeable 12V battery, has the capability to access the World Wide Web, has readily available offline content, via established partnerships, and has an extremely easy to use interface. The device is contained in a single unit with replaceable external components (i.e. keyboard, remotes), consumes less than 100 W of power, and will cost only USD 300. The system is about the size of a shoebox. As seen in the figure, the existing Looma system functional parts can be listed as:

#### 3.1 Looma and Wand Details

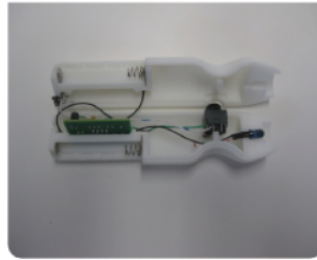
- 300 lumens projector
- Internet Connected
- Wireless Wand control from front of room
- Audio output for large room setting
- Rechargeable 12V battery (8Hrs per charge)
- Computer: Pandaboard
- External ports: 1 Ethernet
- Custom power supply (12V in 5V, 12V, 19.5V out)
- Hacked Wii IR Camera and Wii wands



Hardware and Electrical parts of Looma system

Wand control descriptions:

- Wand shown in the figure is a 3D printed IR light source
- The wand design uses 555 timer to turn the IR light source on/off at a pre-determined frequency such that the blink can be interpreted as a click on the device
- Current design uses Nintendo Wii IR camera, which are scavenged from Wii motes



3D printed hacked Wii Remote wand

### 3.2 System Operation Description

The existing system uses Wii Remote to interact in the audio-visual device. This system uses an Wii IR camera which picks up the infrared light source, and tracks it using its MOT (Multi-Object Tracking) processor present in the camera chip as shown in the figure. Wii IR Camera gives out a pre-set I2C address so the interface board needs an I2C multiplexer. To convert the I2C output to serial, and to demultiplex the signals, the system has been using an FPGA board.

### 3.3 Problem Statement

However, some of the system issues which can not be ignored are:

- **Drawing in the Whiteboard:** While drawing in the LOOMA software, the tracking of the IR led is very unreliable and the drawings are not made as expected.

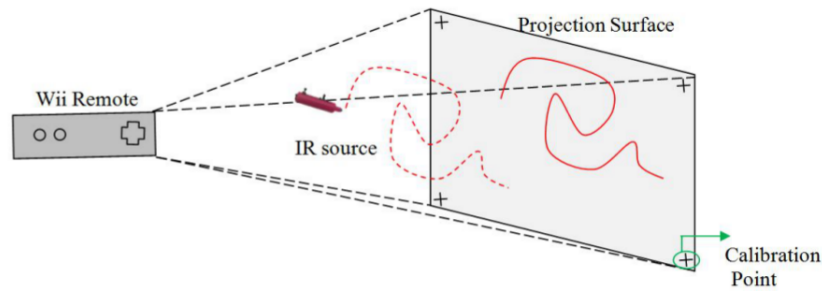


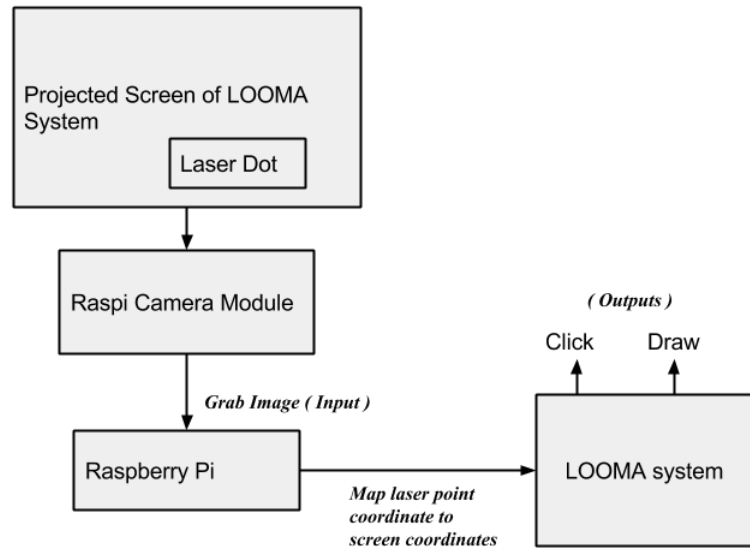
Illustration of how the movement of infrared is picked up by the Wiimote

- **Use of Pre-built MOT processor:** There is the use of inbuilt MOT processor in the camera chip, which does part of the image processing and tracking and sending coordinates. In the case of errors, its response can not be modified to remove errors which makes it inefficient
- **Errors not detectable:** Since the Wiimote used IR to move the mouse, the location of the IR on the screen is not visible with the naked eye thus making errors not detectable
- **Unavailability in the market:** Wiimote IR camera chip being used in the system can not be readily bought in the open market which makes it difficult to mass produce the system
- **Coverage and Difficult to Interact:** Wii wand is not interactive enough and works effectively only within the projected screen space which makes interacting with the screen difficult and not intuitive

In short, hacked Wii Remotes along with the FPGA board has been adding up to the cost significantly. In addition, the system is not very stable and is inefficient to use as points aforementioned, thereby leading to the research on alternative vision-based HCIs and our project is an effort to get a prototype of one such alternative.



## 4 System Block Diagram

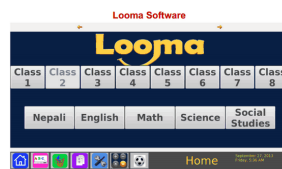


Block Diagram of the working of our system

### 4.1 System Block Explanation

#### 4.1.1 Projected Screen of Looma

The Looma software runs on Looma hardware or Ubuntu. The display resolution of the screen is given by  $X = 1280$  and  $Y = 768$ , and the display width is 60. Once the device is up, there are options of whether to use the wand or not use the wand. In the projected screen of Looma, there is a custom made user interface for the teaching materials for the students as seen in Figure 6 below.



Looma Software

### 4.1.2 Raspi Camera Module

Raspberry Pi Camera Board Module supports a full HD (High Definition) video streaming at fps of 30 with its 5 megapixel native resolution, and the sensor capable of 2592-1944 pixel static images. Raspberry Pi Camera Board Module was opted instead of normal USB web camera.

The software of Raspberry Pi utilizes Rpi GPU (Graphical Processor Unit) when using Raspi Camera Module. So for example encoding *h.264*, video has low impact on the CPU (Central Processor Unit) usage. Also, it has an excellent resolution of 5 Megapixels, which is higher than most USB webcams. It also has an excellent daytime image quality. If we use USB Webcam, it will have a very slow frame rate video and the CPU usage will be quite high. RPi does not have enough CPU horsepower to do higher frame rates, resolution and advanced video compression.

### 4.1.3 Using Picamera

Raspberry Pi camera module will use *Picamera 1.2* by Dave Hughes, recently released on February 06, 2014. This is a pure Python interace to the camera module. Having this interface is a boon since, up until now, the command line syntax needed to be utilized for taking still images and video taking and any changes to the camera's intrinsic features would have required a restart of the camera interfaces and Rpi.

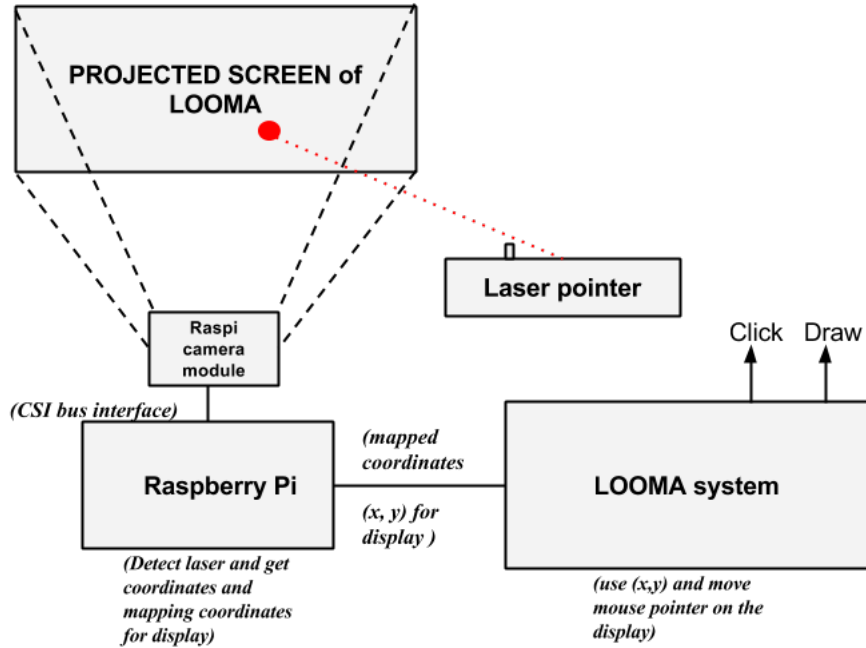
### 4.1.4 Raspberry Pi

Raspberry Pi is a single-board computer with processor, memory, I/O ports and many more features, which together make it a functional computer for a wide range of applications in robotics. It is so simple than any logical person can program it, even it is for the first time when you work with a single-board computer. The ARM powered minicomputer is a platform with enormous possibilities and powerful enough to run many of the same programs as computer. Raspberry Pi serves as a wonderful platform for computer vision algorithms given its size, wonderful camera board and portability. Using the Picamera module, we took raw byte streams of the projected screen, and converted them to OpenCV (Open Computer Vision) object before doing further image processing algorithms.

## 4.2 System Design and Description

As seen in the figures, the camera will be taking in the images of Looma continually and the raspberry pi will be applying the laser detection and click

action detection algorithm on the images. The image processing part will be done in the raspberry pi. Initially, the projected screen must be initially checked for any distortion and warpsness. Depending on the distortion, linear and non-linear screen coordinates mapping matrix will be used. If no such distortion is present, simple linear transformation matrix will be able to map the coordinates to move the mouse pointer accordingly. However, for distortion filled projected screen, non-linear mapping transformation need to be analysed and assessed. Once the mapping matrix has been understood, the raspberry pi takes in the images and works on the projected screen extracted as the region of interest. Now, all the image processing algorithms will be analysed within this ROI (Region of Interest).



Proposed Block Diagram

Next task is to understand the actions of the laser pointer and obtain the coordinates of it and feed it to the Looma System. The actions of the laser pointer are either simple pointing or clicking. Simple pointing will be based on the tracking of the laser pointer in the projected screen and updating the mouse pointer. Clicking of the laser pointer will be understood by the software when the laser pointer is turned on and off for a certain number of times. This on and off pattern of the laser pointer is checked in a number of frames on the same location in the image. This will be implemented as a

software click and the mouse click action is done in the Looma software.

In the Looma System, we will have the system taking in our actions and coordinates and updating its mouse pointer to do the desired mouse actions. Raspberry Pi will be connected to the Looma system through the USB to Serial Port converter wire, with the USB being the part of Raspberry Pi and the Serial Port being the part of the Looma System.

## 5 Works Accomplished in the Seventh Semester

### 5.1 Detection of Laser Pointer and Mouse Movements

Integrating the above the codes of laser pointer detection and mouse handler into one, we were able to move the mouse pointer as per the laser pointer coordinates. Assuming that the camera captures the image from the wall without distortion, a simple linear transformation was carried out, where the laser pointer position in the image taken was mapped to the display resolution of our computer.

From the camera's capture properties, we have extracted the width and height of the image being captured. A simple scalex and scaley have been derived with 1366 and 766 being the display resolution of our sample display device. Once the laser pointer coordinates are found, the transformation is carried out and the mouse pointer coordinates is updated simultaneously so as the laser pointer moves, the mouse pointer coordinates gets updated and it moves with the laser motion.

1. **HSV segmentation:** In the process to detect the laser pointer we used the HSV segmentation and performed histogram analysis and obtained results as shown in Figures 11 and 12 below. The basic steps followed were:

- Capture the frame at each instant of time.
- As the captured image in opencv is in RGB (Red Green Blue) form as default , we convert it into HSV value.
- From the HSV value we made the value of the red color of the pointer as the default such that it can be used for further operations.

*Why is HSV used instead of RGB value?* HSV separates the color component and the image intensity because of which we can operate on each image intensity as per our requirement without changing the color information which plays a vital role in color detection. RGB is the way computers treats color, and HSV try to capture the components of the way humans perceive color.

2. **Masking the image:** We obtained the threshold of the laser pointers color. Using the threshold range, the mask was created. The image and the masked image were operated with bitwise AND operation. Now, only the laser pointer remains in the image, remaining section is masked as black.

3. **Canny Edge detection:** For the edge detection we used canny edge detection algorithm. It is a multi-stage algorithm and we will go through the following stages:

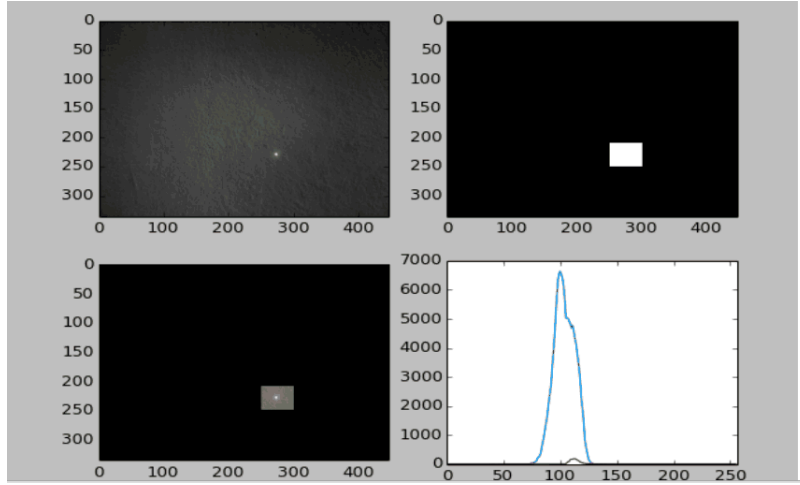
- (a) **Noise Reduction** Since edge detection is susceptible to noise in the image, first step is to remove the noise in the image with a 5x5 Gaussian filter.
- (b) **Finding Intensity Gradient of the Image** Smoothed image is then filtered with a Sobel kernel in both horizontal and vertical direction to get first derivative in horizontal direction ( $G_x$ ) and vertical direction ( $G_y$ ). From these two images, we can find edge gradient and direction for each pixel as follows:

$$\begin{aligned} \text{Edge\_Gradient } (G) &= \sqrt{G_x^2 + G_y^2} \\ \text{Angle } (\theta) &= \tan^{-1} \left( \frac{G_y}{G_x} \right) \end{aligned}$$

Intensity Gradient and Direction

Gradient direction is always perpendicular to edges. It is rounded to one of four angles representing vertical, horizontal and two diagonal directions.

- (c) **Non-maximum Suppression** After getting gradient magnitude and direction, a full scan of image is done to remove any unwanted pixels which may not constitute the edge. For this, at every pixel, pixel is checked if it is a local maximum in its neighborhood in the direction of gradient.
  - (d) **Hysteresis Thresholding** This stage decides which are all edges are really edges and which are not. For this, we need two threshold values, minVal and maxVal. Any edges with intensity gradient more than maxVal are sure to be edges and those below minVal are sure to be non-edges, so discarded. Those who lie between these two thresholds are classified edges or non-edges based on their connectivity. If they are connected to sure-edge pixels, they are considered to be part of edges. Otherwise, they are also discarded.
4. **Contours:** We found the contours of the laser, and its histogram plot was analysed as shown in the figure below.



Histogram Analysis of Laser Pointer

## 5.2 Dynamic Exposure Correction

Correct exposure configuration of the camera can improve the color detection of laser spot. In Raspberry Pi, dynamic exposure correction was done as per need by modifying its camera capture property, which is its camera exposure compensation parameter. Here, the projected screen was captured, and the camera exposure value were modified until color of laser spot was the brightest color in the input image. On every frame captured, the average of the value channel of the image was compared with the preset threshold for seeing laser spot. If the average of the value channel was greater than this threshold, the camera exposure level was decreased accordingly.

## 5.3 Porting of Code to Raspberry Pi

All the codes run and tested in our computer using inbuilt webcam was ported to Raspberry Pi and run using picamera module for using the Raspberry Pi Camera Board Module.

## 5.4 Design Calculations of 555 timer

A simple led circuit is used for the pointing purpose in the wand. To avoid leaving the wand on at all times, a circuit break can be introduced using a simple one-way switch. However, for the clicking purpose, the led circuit must be modified for the system to understand and implement the click. One of the method to signify a click is to switch off the led for a short time but this leads to misintended clicks whenever the led is out of sight of

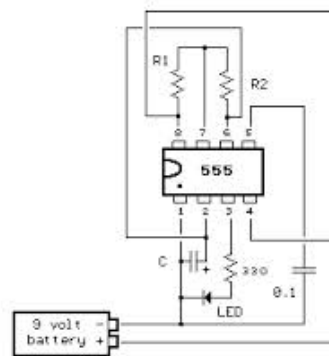
the system. Thus, in order to avoid such errors, the led is made to blink for a specific number of times at a predetermined frequency which is understood by the system as a click. In order to implement this, a flashing led circuit was designed using a 555 timer. The 555 timer is operated in astable mode and outputs rectangular pulses of certain frequency. The frequency is determined by the values of components kept in between the pins of the timer.

The values of R1, R2 and C are set to obtain the required positive time interval(T1), negative time interval(T2) and frequency(f). The calculations are made as follows:

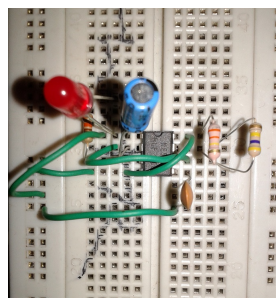
$$T1 \text{ (ms)} = 0.693 * (R1 + R2) * C$$

$$T2 \text{ (ms)} = 0.693 * R2 * C$$

$$\text{Frequency (KHz)} = 1.44 / ((R1 + R2 + R2) * C)$$



555 Timer Calculation



Circuit Test on breadboard

The values of the resistors and capacitor were varied until the required frequency and blinking intervals were obtained. By observation, we found out that R1 should be greater than 1K and C must be greater than 0.0005 uf for a proper working circuit.



## 6 Works Accomplished in the Eighth Semester

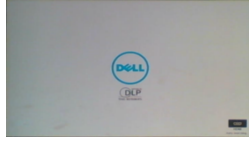
### 6.1 Calibration of Projected Screen

The image projected on the screen is not equal to the size of the computer image, but all the operations and analysis are to be performed with respect to the computer display(or camera frame). So for the calibration of the projector image, we need to map the point in the projector slide to pixel on the camera frame such that all the operations are done on the camera frame. For this mapping purpose, single projective transform is used as the matrix. Our main calibration is done on the basis of the linear mapping which works on the undistorted straight projection screen. However, sometimes, the projected screen in the image has some perspective distortion. Our projected screen is a quadrilateral. The actual four points of the projected screen is used as destination and perspective transform is applied which finds out the transformation matrix for corners of the quadrilateral to the destination coordinates. The following algorithm was implemented in recovering the distorted quadrilateral object.

1. Canny Edge Detection to obtain the edge map
2. Contours Detection to obtain the parent contour
3. Get the four corners of the parent contour
4. Get the centroid of the four corners
5. Determination of Top Left(TL), Top Right(TR), Bottom Left(BL), Bottom Right(BR) points
6. Apply the perspective transformation to a destination coordinates



Distorted Projected Screen



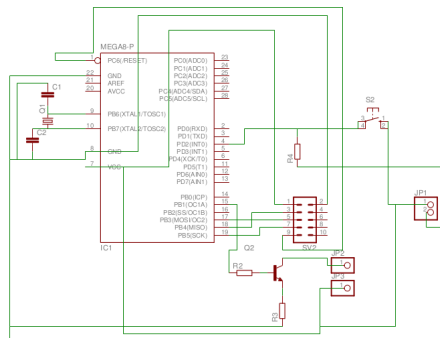
Perspective Transform Correction

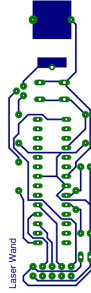
## 6.2 Finalization of Laser Detection Algorithm

In the last semester, the laser pointer had been detected using Hue-Saturation-Value(HSV) color based segmentation for red color. However, the projected screen comprises of many bright red regions which may be perceived as white by the camera. So, the exposure correction of the camera was done dynamically based on the intensity of image taken, such that the only bright spot on the screen is the laser spot. We have applied the HSV segmentation for white color in the Value range of 200-255, and its coordinates are obtained.

## 6.3 Microcontroller to generate laser pulses

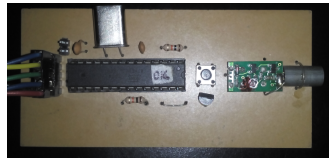
We used an avr microcontroller instead of 555 timer so that we would not need to work with resistor values and capacitors to change the frequencies and duty cycles intermittently. Microcontroller allows us to change the values on the fly and a simple burn would allow us to change the circuit behaviour in a jiffy. Below, is the schematic of our circuit. We have implemented a circuit, that keeps laser on once power is turned on, and when the push button is pressed, it sends out a pulse at a frequency set in the microcontroller. The schematic diagram of our circuit can be followed below, and the schematic as well.





Laser circuit eagle design diagram

The laser circuit has been programmed such that it produces a 6Hz pulses when the push button is pressed, else it stays on. The target frequency has been set to 6Hz. The prescaler used is 64. The top value for pulse width modulation(PWM) is obtained as,  $\text{clock frequency}/\text{target frequency} \times \text{scaler} - 1$ . For a 20% duty cycle, the on time of the laser is set to 20% of the top value and off time is set to the 80% of the top value. We have kept the push button switch in the interrupt pin, so that when the button is pushed, interrupt is obtained and the subsequent Interrupt Service Routine(ISR) can be run. This ISR does the work of producing PWM pulses of 20% duty cycle. The diagram below is our working circuit.



Laser circuit front



Laser circuit back

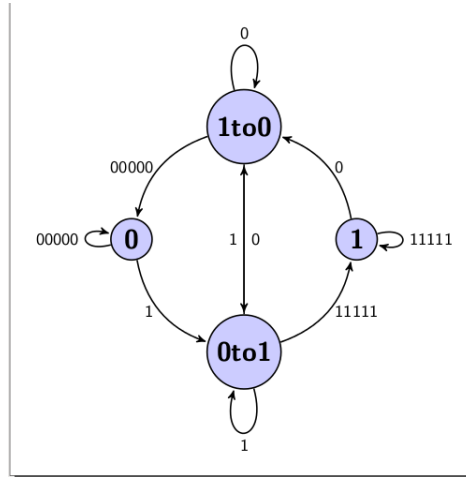
## 6.4 Finalized algorithm for click and drag

We constructed a State Diagram with the following states of laser.

1. Laser-off or 0
2. Laser-on or 1
3. Toggle-0to1 or 0to1
4. Toggle-1to0 or 1to0

**Laser-off** is a state of continual five or more off laser states, which occurs when the laser is not detected. **Laser-on** is a state of continual five or more on laser states, when the laser is detected. **Toggle-0to1** occurs when laser state toggles from off state to on state, whereas in **Toggle-1to0**, laser state toggles from on state to off state.

At all times, when laser is in the on state, the mouse pointer continuously moves. When laser state toggles, either from laser on to off or vice-versa, the mouse pointer is pressed. And when this mouse pointer is moving and also if the mouse pointer is in the toggled state, then the mouse starts dragging. This continues until a laser-off state or laser-on state occur, which is interpreted as a mouse release. Similarly, at every toggle state, mouse down occurs and if laser-on or laser-off state is immediately succeeded to this event, then mouse is released. This will be interpreted as click.



State Transition Diagram of Algorithm

## 6.5 Communication between computer and RPi

Currently, the Looma software is obtaining the mouse coordinates through its serial port from the FPGA board. However, we implemented a protocol

for communication using Transmission Control Protocol(TCP) via ethernet to communicate the mouse actions and coordinates to the computer. The Looma board was made a server, and the Raspberry Pi board a client. The client sends the coordinates of the detected laser spot, and the actions associated with it. Accordingly, the server which is listening to the client, receives the coordinates and their respective actions.

## **6.6 Documentation**

All the researches and case studies conducted will be documented using standard Latex documenting software.

## **7 Works Remaining in the Eighth Semester**

### **7.1 Effective Algorithm implementation**

The algorithm has been finalized and has been implemented in code but needs proper testing and evaluation in order to make it work effectively in Looma board.

### **7.2 Using the mouse library**

The mouse coordinates as well as the actions interpreted need to be fed to the mouse library for the appropriate mouse movements in the Looma board. After the client sends in the coordinates and the actions to the server present in the Looma board, the server parses the data and uses it to run the appropriate library functions.

### **7.3 Integration of the system**

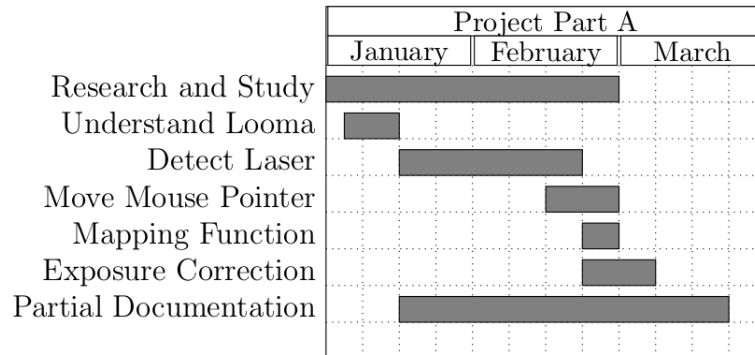
All the parts need to work together, to produce an effective interactive projector system. This can be accomplished once the algorithm is tested well, and implemented together with the mouse library. Also, all the mouse positions need to occur by transforming into the Looma board's coordinates system.

### **7.4 Documentation**

All the remaining works and progresses will be documented using Latex, the documentation software.

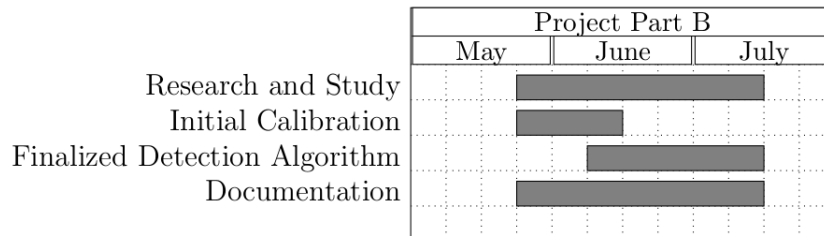
## 8 Project Timeline

### 8.1 Works Completed Timeline 7th Semester



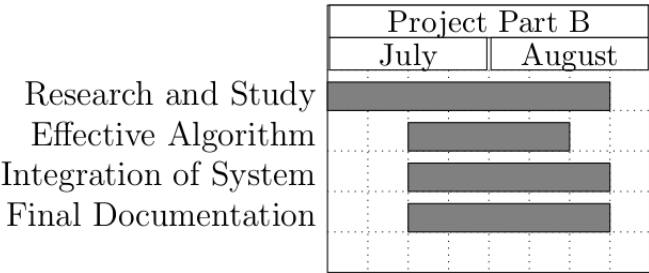
Works Completed Timeline 7th Semester

### 8.2 Works Completed Timeline 8th Semester



Works Completed Timeline 8th Semester

8.3 Works Remaining Timeline 8th Semester



Works Remaining Timeline 8th Semester



## 9 Bibliography

1. Kirstein and Muller, Interaction with a Projection Screen Using a Camera-Tracked Laser Pointer, University of Dortmund, Germany
2. Mikawa, M., Morimoto, Y., Tanaka, K.: Guidance method using laser pointer and gestures for librarian robot, IEEE-RO-MAN, September, 2010
3. Johnny Chung Lee, Hacking the Nintendo Wii Remote, Carnegie Mellon University, IEEE-CS, 2008
4. Kent L. Norman, Kirk D. Norman, University of Maryland, Comparison of Relative Versus Absolute Pointing Device2, May, 2010
5. Matej MESKO, Stefan TOTH, Laser Spot Detection, University of Zilina, Faculty of Management Science and Informatics (2013)

## 10 References

- [www.python.org](http://www.python.org)
- [www.raspberrypi.org](http://www.raspberrypi.org)
- [www.simplecv.org](http://www.simplecv.org)
- [www.villagetechnsolutions.org/](http://www.villagetechnsolutions.org/)
- [www.docs.opencv.org/](http://www.docs.opencv.org/)