•Pointex•
School of Engineering Science • Burnaby, BC • V5A 1S6
ensc440-group-16@sfu.ca

March 8, 2007

Lakshman One
School of Engineering Science
Simon Fraser University
Burnaby, British Columbia
Canada

Re: ENSC 440 Design Specifications for computer input device

Dear Dr. One:

The attached document, *Design Specifications for a Handheld Computer Pointing Device*, outlines the design specifications for ENSC 440. We are in the process of designing a handheld computer pointing device that moves the mouse cursor as the device moves in the corresponding direction, which enables users to operate in situations where flat surfaces are not available.

The objective of this document is to list detailed design parameters for which our device will operate. This design specifications document will include relevant information that applies to be met by the end of April. This document will be used as a reference for designers due to the high level of detail included.

Pointex consists of 5 motivated and dedicated senior SFU Engineering Science students: Frank Chen, Donavan Ho Sui, Randall Lim, Jeff Wong, and Kevin Yang. We can be contacted by e-mail at *ensc440-group-16@sfu.ca*.

Sincerely,

Frank Chen
CEO
Pointex

Enclosure: *Functional Specification for a Handheld Computer Pointing Device*

Design Specifications for Handheld Computer Pointing Device

Project Team:
Frank Chen
Donavan Ho Sui
Randall Lim
Jeff Wong
Kevin Yang

Contact Person:
Frank Chen (*fpc@sfu.ca*)

Submitted to:
Dr. Lakshman One – ENSC 440
Steve Whitmore – ENSC 305
School of Engineering Science
Simon Fraser University

Issued date: March 5th, 2007
Revision: 1.0

Executive Summary

The SmartPoint handheld computer input device is directed towards a market of computer users who wish to operate the mouse with precise control, such as graphic arts users, CAD designers, and artists. The controls of the mouse cursor that SmartPoint will achieve have not yet been met by any other present day mouse input devices. Therefore, it could be a benchmark for future technology whereby other products may use our system as a basis for operation.

The development of SmartPoint will occur in multiple stages. The beginning stages will include development of the basic foundation our product such as the underlying systems. Subsequent stages will use this underlying system and make further advancements or slightly alter the original design.

After the first stage of development, SmartPoint will:

- Move the mouse cursor corresponding to the movements of the user.
- Have the “left-click” and “right-click” functionality.
- Have a Lithium Ion rechargeable battery that comes with charger
- Be wireless.

Subsequent stages may include:

- A new gamer-intuitive mouse operating system.
- Wheel scrolling function.
- Indoor and Outdoor operating modes.
- Additional add-on buttons for easy web surfing.
- Implementing tilt sensors.

We estimate that the first stage of development for SmartPoint will be completed by April 2007.

Table of contents

List of Tables

List of Figures

List of Acronyms

CAD: Computer Aided Design

FPS: Frames per Second

GUI: Graphical User Interface

IPL: Intel Image Processing Library

IR: Infra-Red

LED: Light Emitting Diode

OpenCV: Open Computer Vision

PC: Personal Computer

USB: Universal Serial Bus

1. Introduction

Studies have shown that people working on laptops with an external mouse are much more productive, faster, and accurate than those working with the laptop's built-in touchpad or trackpoint. Unfortunately for people working in a predominantly outdoor field environment, such as geologists and biologists, flat surfaces required by mice are not readily available.

The objective of our project is to develop an alternative pointing device, SmartPoint, that these people can use without having to resort to their built-in pointing devices and compromise their productivity. The general idea is a handheld device that navigates the mouse cursor around the screen by pointing the device at the same location, much like video game light pistols.

In addition to workplace benefits, SmartPoint can also augment the computer gaming experience by providing a more immersive way of interacting with the game. Instead of using the traditional mouse to control the movements of an onscreen weapon, the user can experience the simulated sensation of holding an actual gun while pointing at the objects on the screen.

1.1 Scope

This document standardizes the specifications to feature in the proof-of-concept and production of SmartPoint models. The list of design specifications for the proof-of-concept model provides the guidelines for our design engineers. Since experience will be gained while developing this device, a vast majority of specifications are supplied for the production device.

1.2 Intended Audience

The primary audiences of this document are design engineers, integration engineers and quality assurance personnel. This document will be the guideline through which all module and system designs must comply. Engineers and executives can use this document to plan, direct, motivate and monitor the development progress. Marketing personnel can use this document to develop advertising materials.

1.3 Referenced Documents

[1] Proposal for a Handheld Computer Pointing Device. Pointex

2. System Requirements

2.1 System Overview

Our system structure has a uni-directional flow of information. The user holds a mouse tracking device in his hand which transmits point reference signals to the webcam positioned above the display monitor. Based on the user's hand movement, the webcam peripheral interprets this transmitted data and moves the mouse pointer appropriately on the display monitor.

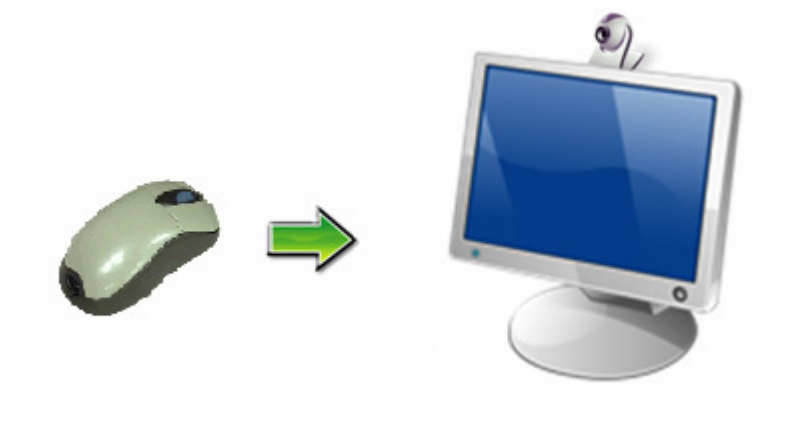

Figure 1 Device System Overview

2.2 Handheld Remote Subsystem

The handheld device subsystem will be small and portable. Motion movements by this device will be identified and tracked by the web camera and software subsystems. This subsystem will also contain buttons that the user will use in order to perform "left click" and "right click" operations, same as any ordinary computer mouse.

The web camera and software subsystem will move the mouse cursor corresponding to the user's hand motions, and is therefore responsible for relaying user input information to the PC. This subsystem emits a traceable reference marker, thus allowing the camera to identify and capture all user data. The data will be processed by the software subsystem. Pressing the "left click" and "right click" buttons will emit different reference markers, which will execute the corresponding clicks on the PC. Using the subsystem in this way allows for the remote to be completely wireless, without the need to send any radio-frequency signals.

2.3 Software Subsystem

The software subsystem is the system process the computer will use to analyze the images captured by the camera. The programs that will be used are Visual Studio C++ and OpenCV. There will be a number of required processes, of which each process will require a portion of the computing power. Each process is essential in analyzing the web camera images correctly and efficiently. The software subsystem will include processes such as: matrix calculations, mouse movement, image processing, and background subtraction.

The user may wish to position the mouse cursor in a way that requires that user to perform the least amount of hand movement. This means the user will be able to tilt the device from a stationary point instead of dragging the device across the surface of the monitor. Therefore, the software subsystem will be required to calculate from the images both the angles and tilt of the device, in order to determine pointing position on the display screen.

Image processing requires background subtraction which enables object tracking in an efficient manner. Background subtraction is a part of the image processing subsystem, whereby it cancels all other background colours except for the colour of the reference marker, thus making the marker tracking process much easier.

After the background subtraction subsystem, there will be a much more simplified image left to be processed. The image processing subsystem includes all modules which analyze images sent from the web camera and determine the distance of reference marker movement. This information is then sent to the mouse movement subsystem. The mouse movement process will take the information of mouse displacement and possible mouse clicks, and communicate these commands to the PC.

2.4 Camera Subsystem

This product will require an input capture device to obtain images, having a minimum defined resolution and frame rate, to be sent to the software subsystem for processing. The resolution and frame rate of the camera subsystem will determine the performance characteristics of the mouse cursor, and also the amount of computing power required to process the images. However, an existing web camera would be sufficient in obtaining the necessary input from the handheld device which will then be sent to the PC for image processing.

3. Physical Remote Pointer Design

This section contains all relevant information pertaining to the design of the pointing device component of the system.

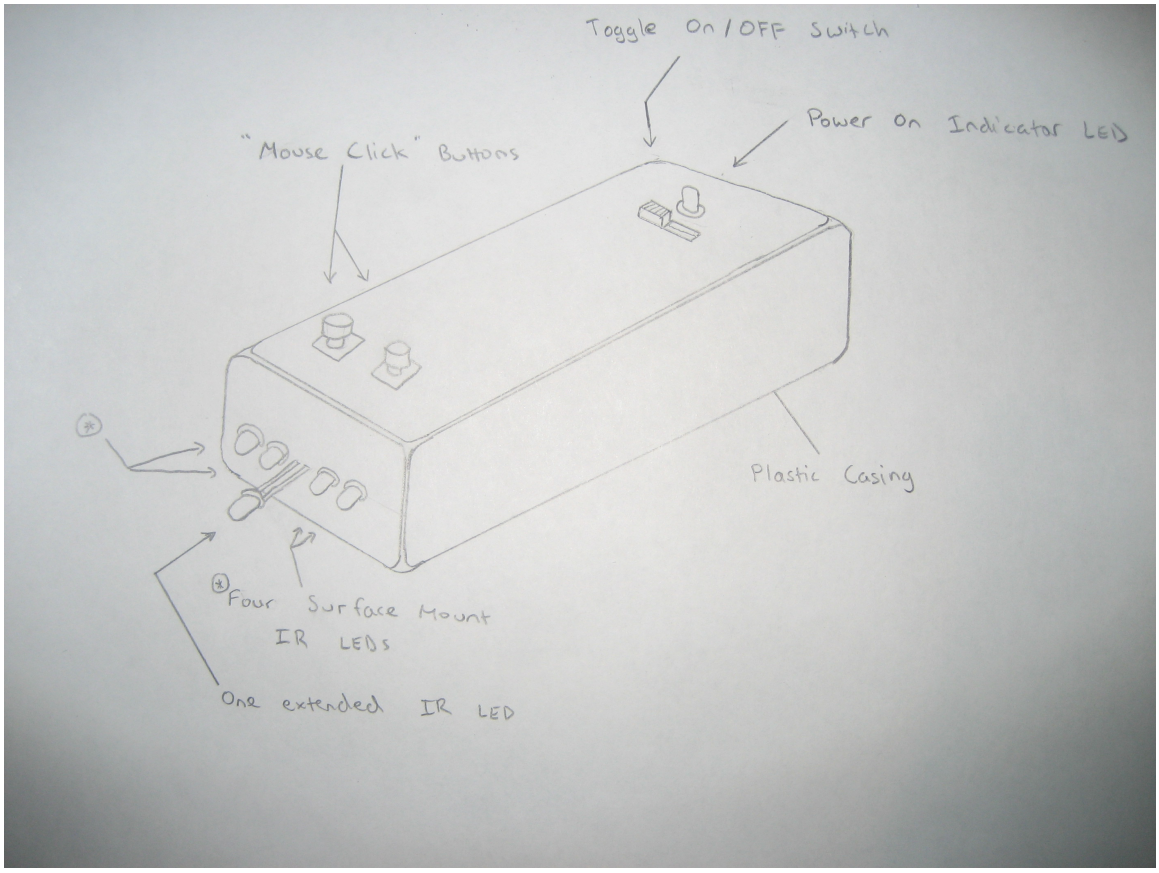

Figure 3.1: Concept diagram of device

3.1 General

The SmartPoint remote pointing device is used as a reference marker for the system, through the use of basic physics. The design we have chosen will use Infra-Red (IR) light emitting diodes (LEDs) as the markers, and the web camera will capture this type of radiation. Without any type of light filtering, a full spectrum radiation (colour) image would be captured by the web camera and post image processing requirements for the computer would be much more significant. Therefore, for simplicity and reduced processing requirements, we have chosen to attach an IR filter to the web camera to filter out all other light radiation including visible light except for IR light. The use of IR LEDs and the IR filter produce usable images for post processing.

One mode of operation which the development team at Pointex will attempt to implement is the “tilt function”, where the user can pivot the pointing device and move the mouse cursor to the corresponding position on the monitor. This function requires multiple reference markers to determine location, as well as a pointing vector. Shown in the figure below, a basic configuration of reference markers was specified so that this function may be possible to implement. A switch will be needed to change the modes of operation.

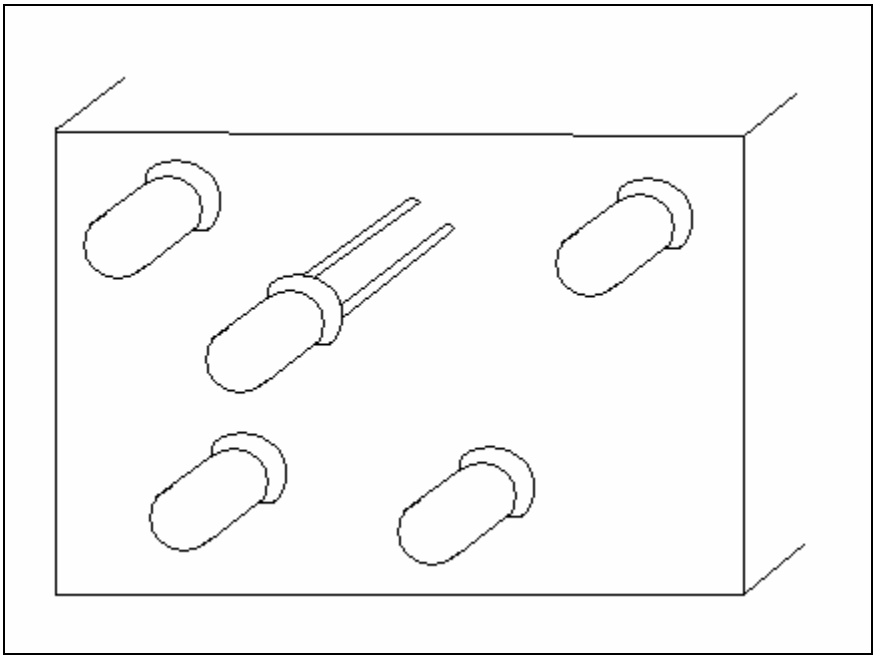

Figure 3.2: Simple diagram to show reference marker layout

This product will have both the “left-click” and “right-click” functions. The design team has specified the use of reference markers as a means of relaying the button clicking functions to the image capture system. Having one extra lit marker will indicate a “left-click” function and two extra lit markers will indicate a “right-click” function. Switches will be needed to apply these signals.

For the prototype versions, power for the device will be implemented through the use of two of either AAA or AA type batteries. The size of the battery will impact the weight and electrical capacity (battery life) of the SmartPoint remote, as well as the size of the casing used to hold the batteries. Further experimentation will be needed to determine the battery type.

Subsequent versions may have rechargeable batteries implemented into the pointing device, where recharging takes place through a USB charging station.

3.2 Enclosure for SmartPoint remote

The enclosure for the pointing device will be a casing made from a hard durable plastic. The plastic enclosure must encase all the required components, and be reasonably safe to the user. A working physical protoype for the material and enclosure can be found from the figure “Prototype 1”.

3.3 Infra-Red LED

Power Rating	
Maximum Voltage	1.6 V
Maximum Current	50 mA
Measured Voltage	1.34 V
Current in series with 100 Ω resistor	1.5 mA

Table 3.1: IR LED Power Rating

For our system, a maximum of five IR LEDs will radiate at a given time.

3.4 Clear Water Red LED (power-on indicator LED)

Power Rating	
Maximum Voltage	1.85 V
Maximum Current	20 mA
Measured Voltage	1.52 V
Current in series with 100 Ω resistor	1.5 mA

Table 3.2: Red LED Power Rating

This LED will be used to indicate when the power for the SmartPoint pointing device is turned on. Since a human eye cannot receive Infra-Red light, an indicator LED is required to determine the operating modes.

3.5 Momentary Push button

The momentary normally open push button has typical characteristics with a maximum current of 1A, which is much greater than our purposes, as well as a negligible voltage drop. These push buttons will be used as the “left-click” and “right-click” functions.

3.6 Two position switch

The double position switch has three terminals for contact and can be used as an on/off switch. This switch may be used to change the modes of the device, or to turn the entire remote on/off. This device has a maximum current rating of 300 mA.

3.7 Battery

This is the current power source for the remote. Future implementations may use a rechargeable battery and a recharging station.

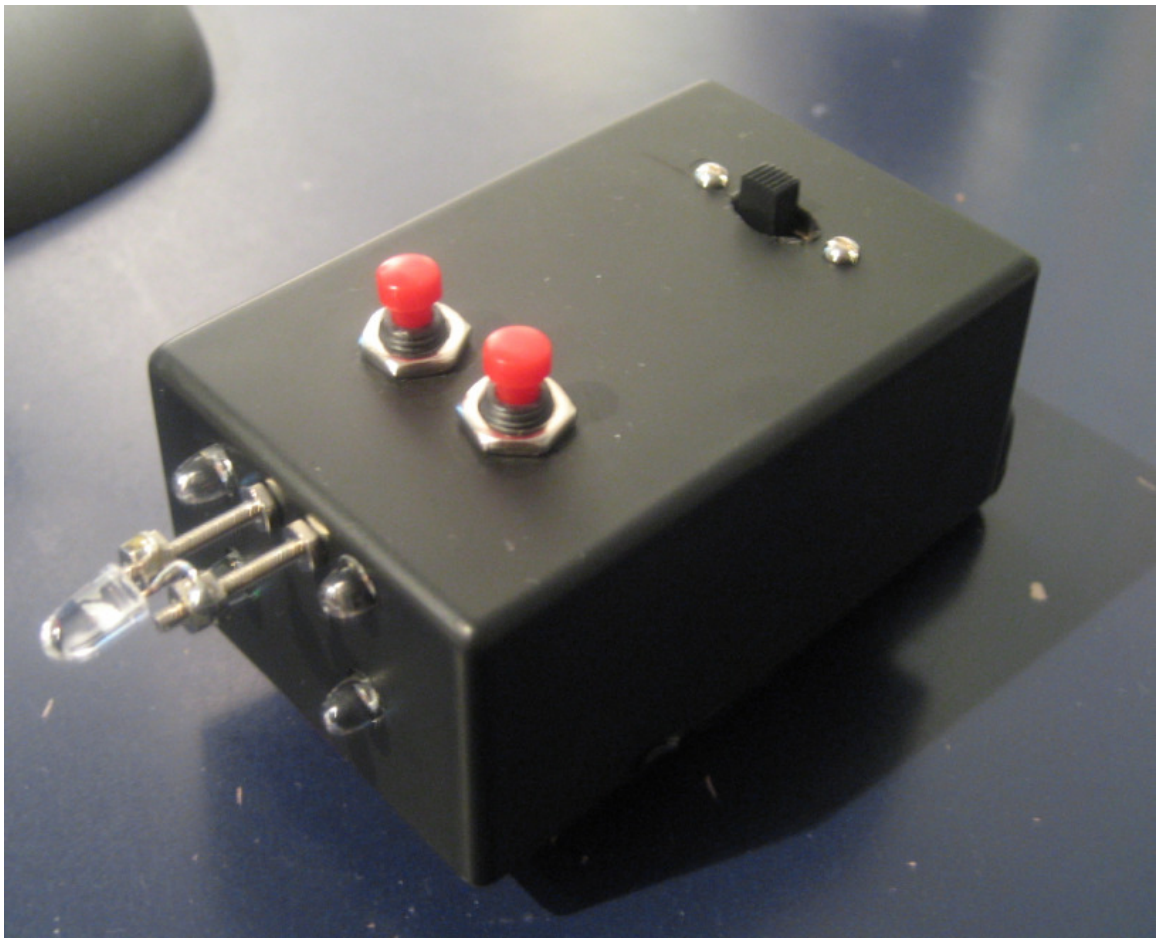

Figure 3.3: First prototype version 1

3.7 Schematic

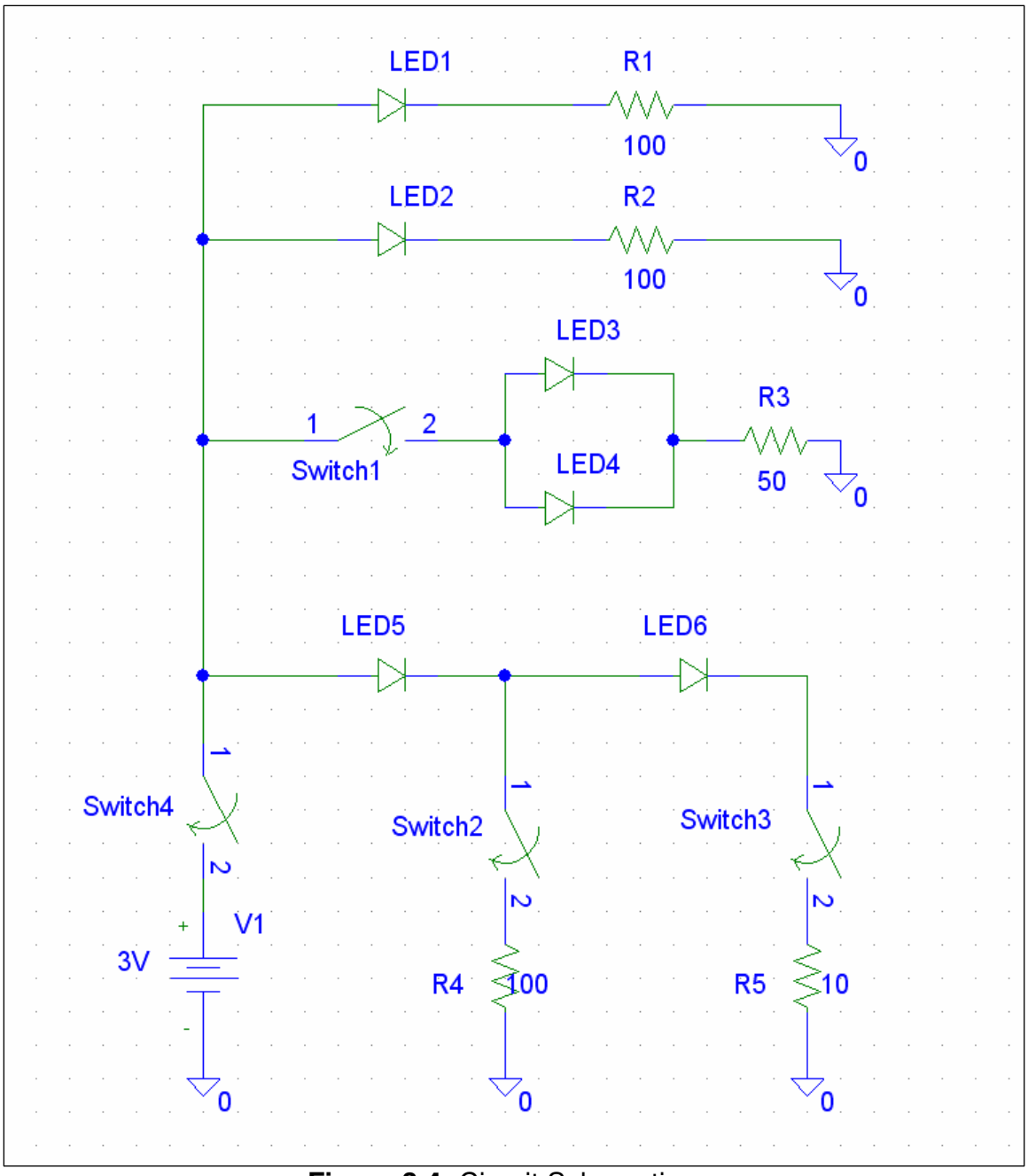

Figure 3.4: Circuit Schematic

4. Software Design

4.1 General

Image processing is used to obtain frame images from the web camera and subsequently derive important information from the images such as the tilt of the device, number of LEDs currently lit up, and position of the device with respect to the x-y coordinates of the frame. Furthermore, various modules such as background subtraction, distance calculation of reference markers movement, and tilt calculations are implemented to properly and accurately control the movement of the mouse cursor.

4.2 Programming Language

The software programming will be coded in C++ language. C++ is an extension of C and offers programmers low-level control over the program with instances

such as pointers, while at the same time allowing expression of abstract ideas at a higher level as compared to assembly languages. C++ is object oriented and allows easy re-use of code, or parts of code through inheritance.

The Visual Basic language, however, is more oriented towards event driven programming and is ideal for creating GUI applications and ActiveX controls and objects. Java on the other hand, is similar to C and C++ as it is also an object-oriented programming language. However, its main fallback is its inability to accommodate low-level controls and is structured towards high level coding. Taking into consideration both the advantages of the various languages, C++ is ideal language based on given the prototype requirements

4.3 C++ Program Library

Open Computer Vision (OpenCV) is a library of programming functions mainly aimed at real time computer vision. The library is compatible with IPL and utilizes Intel Integrated Performance Primitives for better performance. OpenCV library is useful in many applications such as object identification, segmentation and recognition, and motion tracking. The code of this project heavily relies on many of OpenCV's functions which allow tracking of the reference markers possible.

4.4 Frame Capture using OpenCV

As part of OpenCV's libraries, it can also communicate with the web camera and create an initialization of the device. Using only a handful of its functions, OpenCV can capture the raw data in every frame taken from the web camera and store in into an image buffer of type IplImage, referred to as IPL image header, which can be used for further processing in the code. Also, using other necessary information obtained through the communication with the web camera such as image width and height in pixels, pixel depth in bits, and placement of origin, OpenCV can read from the image buffer and display the real-time image in a pop-up window. The frame rate of the image buffer is limited by the frame rate of the web camera hardware.

4.5 Background Subtraction

Background images and/or lighting can cause incorrect identification of the reference markers and thus distorting the proper movements of the mouse pointer. Therefore the purpose of background subtraction is to remove all static background lighting that could interfere with the prototype's reference markers. An effective way of separate the unwanted background from the prototype's markers is to implement motion detection using the captured frames of the web

camera. This is done by taking the previously captured image frame and subtracting it to the current frame. The resulting frame now only shows the differential movement while the static background gets blacked out.

If differential movement is sensed after background subtraction is applied, it is safe to assume that this movement is caused from the displacement of the reference markers. Each pixel in the frame is accessed and read to determine what the RGB value is per pixel. The position of the reference marker is determined by comparing the pixel value with a threshold value. Thus, if the pixel value is greater than the threshold value, the pixel can be considered a brightly-lit pixel which captured the reference marker. The x-y coordinates of this pixel is stored in two integer arrays.

Using the integer arrays, a minimum and maximum value is found in each array. These coordinates are used to draw a box that encompasses all of the reference markers. An exception to consider when using background subtraction would be if the user points the device at the web camera while holding it in a static position. The end result from background subtraction would be that the reference markers would disappear as it is also a static image. To take this into account, every pixel that is within the box area is re-written with the pixel values taken from the original frame prior to background subtraction, thus eliminating the problem of having static reference markers.

Figure 4.1: Image prior to background subtraction

Figure 4.2: Image after background subtraction is applied

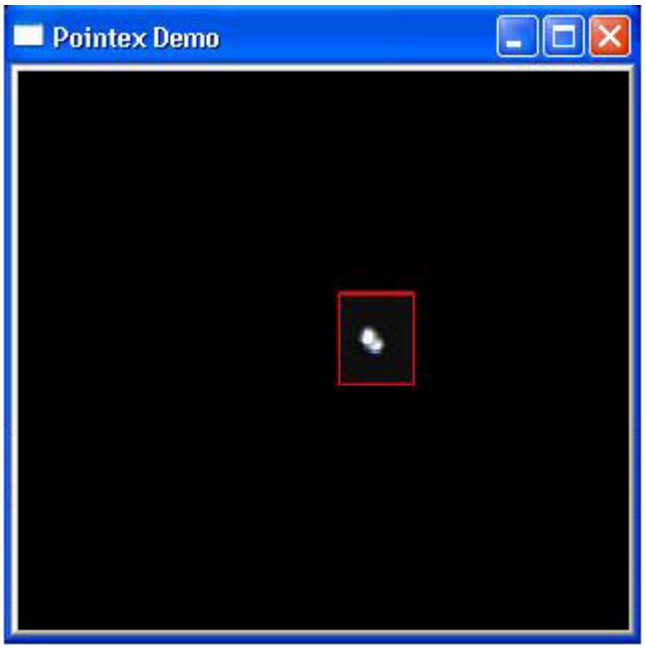

Figure 4.3: Image tracking with background subtraction

4.6 Accessing Pixel for Color

Accessing each pixel element of every captured screenshot will be done using OpenCV's built-in CV_IMAGE_ELEM function. With specified coordinates provided into its parameters, the function returns a Unicode character that contains the RGB colour contribution of the targeted pixel that range from 0 to

255 for each colour. Using these values, we are able to directly measure quantities such as brightness and colour balance on any part of the image. This technique forms the basis of all our image processing.

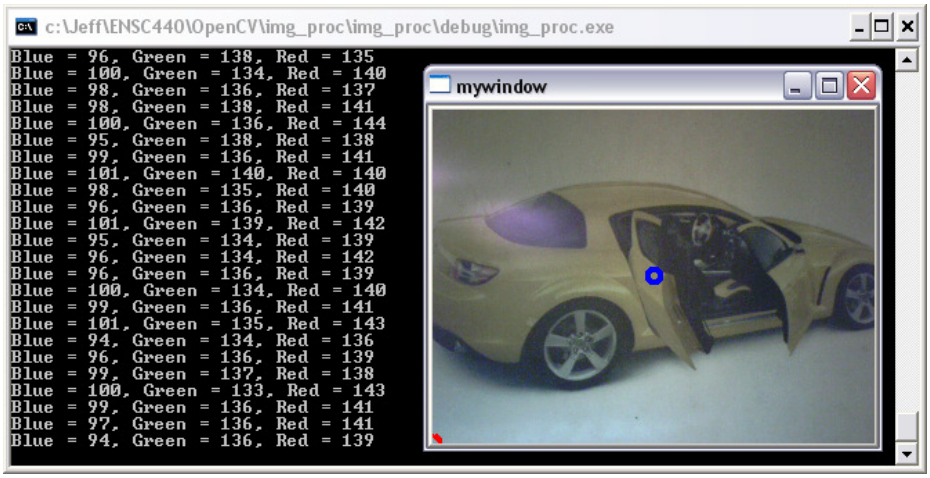

Figure 4.4: The colour of the pixel at the center of the blue circle is analyzed

4.7 Bright Points Search

We were able to access the each pixel for its RGB color contribution. With the film filter and background subtraction, the sum of color contribution of an IR led will have a value of at least 500. Therefore, a for-loop is used to go through every single pixel that the web camera captures and search for bright points.

Figure 4.5: Bright Point Detection (Green Dot)

4.8 Center of Each Bright Points

Since each IR LED will most likely have more than just 1 pixel over 500-color contribution, a set of bright pixels can be grouped and then having taken the average of their x-y coordinates to determine the center position of the bright point to increase accuracy.

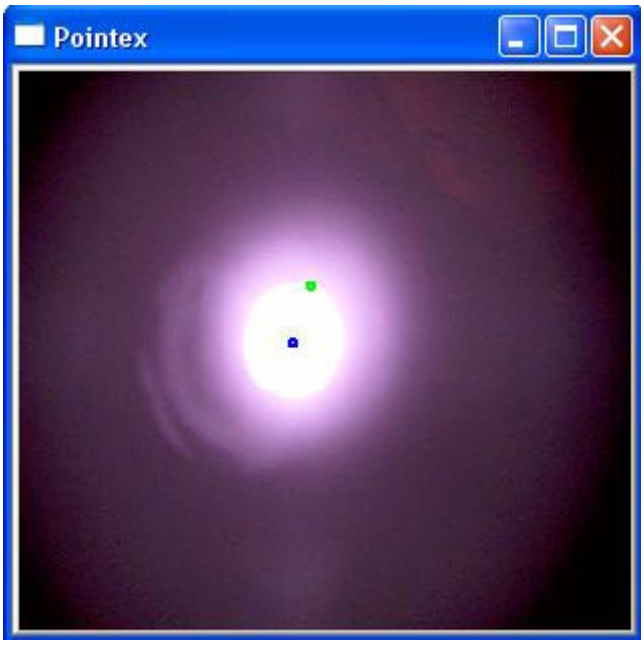

Figure 4.6: Bright Point Center Detection (Blue Dot)

4.9 Mouse Actions

C++ contains a library that will communicate will the mouse. It can move the mouse by a certain number of pixels in the x-direction and y-direction. It can also simulate left and right mouse button click. These will be integrated with the reference point tracking.

4.10 Multiple Bright Point Search (MBPS)

This is an extension to the Bright Point Search. We are to implement multiple scans to pick up the locations where the IR LED appears on the captured image. Since the LED may take up more than 1 pixel in the captured image, the scan isolates a grid around the located point to separate 1 LED from the next. The size of the grid perimeter depends on the distance at which the user operates the device (Section 4.11 - Distance Detection). Doing the multiple scanning this way would solve the problem of detecting the same LED more than once. With this

algorithm we can implement the tilt of the mouse which requires multiple LED recognition prior to moving the mouse cursor. We are also able to incorporate this with the mouse clicks using the libraries provided in OpenCV.

4.11 Distance Detection

In order to detect the distance from the web camera and the handheld device, we will implement the detection of the number of bright pixels. The closer the user holds the web camera, the more pixels each LED will take up in the captured frame. We will take a range of the number of bright pixels detected and correlate that to the distance from the web camera.

4.12 Mouse Clicks

The Left and Right mouse click function is incorporated using the library provided by OpenCV and the Multiple Bright Point Search portion of the code. In order to simulate a left mouse click, the MBPS code needs to detect the color contribution of an additional IR LED that is designated for a Left click (aside from the other 3 motion IR LEDs). To simulate a Right mouse click the MBPS code needs to scan for 2 additional IR LEDs that are activated when the Right click button is pressed. Outside of the Left mouse click conditions, we will simulate a constant release of the left mouse click. For the Right mouse click, it is activated only on the release of the click so we can simply ignore the conditions to simulate the right button to be pressed down.

In order to separate the mouse simulation LEDs from our mouse cursor LED, we need to parameterize the displacement of each LED detection. We do this to avoid losing track of our initial reference point when new LEDs are detected. Because multiple bright points are present in a captured image, it is possible that the current x-y coordinates of the LED will jump to the location of the second lit LED when it is introduced in the next image frame. Implementing a displacement restriction for each LED will resolve this problem.

4.13 Multithreading

To increase overall throughput of the application, we will implement multiple threads into the application process to enhance the speed of the more critical functions such as image processing and buffer updating. Threads are smaller processes within the main software process that run periodically and independently.

5. Web Camera Subsystem

5.1 Frame rate

The camera's output frame rate is crucial for smooth and accurate operation of SmartPoint. At this point, the absolute minimum frame rate the web camera must be capable of is 10 FPS.

5.2 Resolution

Resolution also plays a critical role in the precision of SmartPoint. At this point, the absolute lowest resolution of the image must be 320 x 240 pixels.

5.3 Field of view

A large field of view allows more space for the user to interact within. At this point, the smallest field of view the web camera must be approximately 60°.

6. Conclusion

This document includes all the functional specifications of the three subsystems that are in the SmartPoint device. The details listed are all that are required to successfully complete and test the construction and operation of the SmartPoint. By April 2007, the SmartPoint will be completed to meet all the specifications listed in this document.

