SEES PROJECT

**Spatial Echolocation Enhancement System**

**Hardware Design Document**

University of Victoria

CENG/ELEC/SENG 399 Fall 2014

Design Team 14

Daniel Faulkner - V00778450 - SENG

Paulo Tabarro - V00810880 - CENG

Jason Lim - V00785426 - SENG

Adalberto Outeiro - V00810849 - ELEC

Rajpal Chauhan - V00762290 - ELEC

# Introduction

This document will describe the hardware specifications for the SEES Project headset peripheral. The headset peripheral is a device worn by the user that holds a sensor used to acquire depth information about the user’s environment. The information gathered by the headset is sent to the system software on the user’s mobile device which then provides appropriate feedback to the user. The topics covered in this document include a description of the headset components, an illustration of the operating principles of the headset sensor, a format specification of the data that is captured, and the minimum operating requirements of the user’s mobile device.

# Headset Components

The SEES headset peripheral will utilize a depth camera based off of a stripped down version of the Microsoft Kinect 2.0 depth camera. This includes an 830nm laser emitter as well as an integrated infrared camera [1]. It also includes a depth camera processing board that is used to convert the measurements from the camera sensor into a depth image. The headset will connect to the user’s mobile device using a regular mobile phone USB-to-Micro-USB cable. This cable will both act as a power source for the headset as well as a communication line to transmit sensor data along. Table 1 summarizes the headset components:

**Table 1:** Headset peripheral components

|  |  |
| --- | --- |
|  | **Description** |
| **Wearable Frame** | A wearable headset housing for the headset components. |
| **Infrared Emitter** | A 830nm laser diode for emitting structured light onto the scene. |
| **Infrared Camera** | A monochrome image sensor with an IR-pass filter of 830 nm. |
| **Depth Camera Processing Board** | A processing board for converting sensor measurements into a depth image and sending it across the USB line. |
| **USB to Micro USB cable** | A USB connector cable for connecting the headset to a mobile device |

# Communication Channel Sensor and Mobile

Despite being a robust device and offering a great part of the functions the project requires, the Kinect has some compatibility and communication issues that need to be addressed. Currently, many projects using Kinect technology outside Xbox consoles are in the Windows environment. Windows clearly has not focused on making their device to be universally compatible, since their documentation approaches only solutions for windows-based machines. Also, Kinect has the need of being supplied by a power outlet, and requires some modifications to make the device USB compatible [2], which imposes some geographical limits to the hardware and mobility progress of the project.

On the other hand, independent projects have shown manners to dealing with this incompatibility and energy dependability. Mobile Kinect, a project that integrates Kinect with an android PC on a stick, innovated running Kinect on a Gumstix Overo Summit board, routed to the mini PC via a separated router unit. This project also uses a battery supply for powering up the devices with some tension filters:

1. LM1085-5V, 5V 3A LDO regulator
2. LD1086-3V3, 3.3V 1.5A LDO regulator
3. USB connector, Molex OTG

# Operating Principles

Due to Kinect being a popular and well supported device, the group decided to make use of Kinect for the project. Kinect is motion sensing input device which enables its users to connect and interact through a natural user interface using gestures and spoken commands. The depth sensor consists of an infrared laser projector combined with a monochrome CMOS Sensor, which captures video data in 3D under indoor ambient light conditions. The sensing range of the depth sensor is adjustable, and Kinect software is capable of automatically calibrating the sensor based on gameplay and the player's physical environment, accommodating for the presence of furniture or other obstacles [4, 5].

With this in mind, it was decided to connect Kinect with mobile phones in order to get images of objects. In order to promote the use of this device across a wide range of domains, it will be necessary to make the circuit diagrams and PCB layouts for the additional circuitry available. The design will only use the front camera circuit board of the Kinect and a second board of the same small size that plugs onto the back of this board [6]. This will connect to the user’s smartphone. The design would work equally well with a Raspberry Pi.

# Sensor Data Format

For depth sensor, the technology that is going to be used is the same as the used in the Kinect. This technology requires an infrared laser projector combined with a monochrome CMOS (complementary metal-oxide semiconductor) sensor. For the SEES purpose, the color VGA camera used in the Kinect is not necessary.

The depth sensor has a 640×480-pixel resolution and runs at 30 FPS (frames per second). While the resolution of the infrared sensor is actually 1280x1024, the depth image provided by the sensor is at a reduced resolution due to bandwidth limitations on the USB connection. In indoor environments, the sensor has an operating range between 0.8m to 4.0m [7]. Being an infrared based sensor, the sensor suffers from infrared saturation when used in outdoor environments under direct sunlight. This can lead to outliers or gaps in the resulting depth image. Prototype testing will determine if this limitation it too constricting, and if changes to the sensor unit need to be made.

The depth data is the distance, in millimeters, to the nearest object at that particular (x, y) coordinate in the depth sensor's field of view. Each frame of the depth data stream is made up of pixels that contain the distance (in millimeters) from the camera plane to the nearest object. An application can use depth data to track a person's motion or identify background objects. The process is quite similar to getting data from the RGB image, but instead of RGB values, it returns distance data. The distance information is converted into an image representing the depth map.

# Mobile Operating Requirements

The processing unit under consideration for the SEES project is a smartphone device running the Android Operating system. The Android Operating System facilitates the software development when applying desired HRTF as it can support running native C code [3].

The processor clock must be fast enough to run the HRTFs as well as give adequate feedback to the user (response no longer than 15ms).

|  |  |
| --- | --- |
| **Figure 1.** Smartphone running Android | Another consideration to make is the battery life of the smartphone while operating the system, and the CPU load the system will place on the device. While the original Kinect unit requires an external power supply to run an onboard motor, the SEES headset will be using a stripped down version of the device utilizing only a camera and emitter. The primary questions here are:   * How long can a smartphone power the system on a single battery charge? * Can the user’s phone run the system concurrently with his or her other applications?   Testing will be required to determine if an additional power source and/or processing board is needed. |

# Conclusion

Microsoft Kinect seems to be an affordable and ready-to-use solution to our approach, but it stills has some gaps that need to filled. Working with the connection in a way to ponder how complex the system can get, how many devices would still make a portable device, and how the energy will be supplied to the system are some of the barriers that the hardware design faces at the moment. Moreover, the infrared sensor and the Kinect’s low usability in sunlight might be a cause for a spatial impediment for the project. Testing will determine whether this sensor will meet the system’s requirements, or if another solution will need to be considered.

# Appendix A - References

[1] OpenKinect. (2011, Feb. 24). *Hardware info - OpenKinect* [Online]. Available: <http://openkinect.org/wiki/Hardware_info>

[2] Instructables. (2012). *Wiring an Xbox Kinect for USB* [Online].

Available: http://www.instructables.com/id/Wiring-an-Xbox-Kinect-for-USB/

[3] A. J. Bianchi and M. Queiroz, “On the performance of real-time DSP on Android devices,” in

*Proc 9th Sound and Music Comp. Conf.,* Copenhagen, Denmark, 2012, p.113-120.

[4] M. Wilson. (2009, Jun. 03). *Testing Project Natal: We Touched the Intangible* [Online].

Available: http://gizmodo.com/5277954/testing-project-natal-we-touched-the-intangible/

[5] S. Totilo. (2009, Jun. 05). *Microsoft: Project Natal Can Support Multiple Players, See Fingers.*

Available: http://kotaku.com/5279531/microsoft-project-natal-can-support

-multiple-players-see-fingers/

[6] M. T. Marshall *et al.,* (2012). *Mobile Kinect* [Online].

Availa[ble: http://big.cs.bris.ac.uk/projects/mobile-kinect](http://big.cs.bris.ac.uk/projects/mobile-kinect)

[7] Microsoft Corporation. (2012). *Kinect Sensor* [Online].

Available: http://msdn.microsoft.com/en-ca/library/hh438998.aspx

# Appendix B - List of Tables

|  |  |  |
| --- | --- | --- |
| ***Figure*** | ***Description*** | ***Page Number*** |
| **Table 1** | Headset peripheral components | 2 |

# Appendix C - List of Figures

|  |  |  |
| --- | --- | --- |
| ***Figure*** | ***Description*** | ***Page Number*** |
| **Figure 1** | Smartphone running Android | 3 |