Summary

This document describes the Spatial Echolocation Enhancement System (SEES) Project and the procedures and designs that went into the development of the project. The goal of the SEES project is to develop a system to aid blind individuals with open world independent navigation. The system uses a novel approach of tagging objects with binaural audio cues that act as a natural extension to the user's ability to echolocate objects in the surrounding environment. The scope of this document primarily focuses on the the design of said system and relevant considerations that went into developing it.

Before beginning development of the project, 3 other navigational aids for blind people - ARGUS FP7, Headlock, and GIST - were evaluated. Review of these projects provided insight into some of the techniques currently employed in navigational assistive technologies as well as where these designs could be improved upon.

Following the design reviews, a set of design requirements and specifications were established for the SEES device. This includes the expected operating characteristics of the system, it's inputs, it's outputs, and the processes to be employed within the system.

Various design considerations had to be weighed and evaluated in developing the final design. These considerations included the design of the SEES peripheral, the sensor type to use in the peripheral, the processing technique to use in generating the binaural audio cues, and the power limitations imposed by the processing device of the system.

In order to evaluate the design decisions chosen, testing was done using a software prototype on a limited user base. Testing highlighted some of the limitations of the current system as well as places where the design could be improved. While there were additional limitations imposed due to the nature of the tests being executed in a virtual environment, participants in the test were able to identify objects and navigate virtual environments relying solely on auditory cues.

Taking into consideration the design evaluations as well as the results of the tests, this report recommends the development of a headset styled device that uses time of flight depth camera to capture information about the user's environment. This device will connect to a mobile phone that performs software based fourier analysis of regular audio signals to generate spatialized audio cues. Finally, this report recommends that additional testing be conducted to analyze the distance and lighting limitations of the chosen infrared based ToF sensors, the power constraints of common mobile devices, and the impact of the generated audio cues on the user's ability to hear regular audio cues in the world around them.

1.0 Introduction

The SEES Project aims to develop a system that can aid visually impaired or blind individuals in the task of navigating the world around them. For many blind individuals, the lack of working vision is an obstacle that limits the individual's freedom, capabilities, and quality of life as they go about their daily lives. Currently, there exists a variety of tools and aids to assist these individuals with daily life navigation such as canes, guide dogs, and human guides; however, many of these aids have limitations that restrict either the usability or effectiveness of the aid. Moreover, navigational aids that depend on a person or guide animal are limited in that the guiding agent must be present with the blind individual. Not only is this a burden for both the guiding agent and the blind individual, but it also compromises the blind person's personal independence.

The SEES device described in this document is a navigational device that acts as a natural augmentation to the user's existing ability to navigate using audio cues from the world around them. By mapping spatialized audio cues to otherwise silent objects in the world around the user, the user is able to identify and navigate around objects using their own natural sense of hearing. Not only does this significantly improve the individual's personal independence in being able to comprehend the world around them on their own, but it is also provides navigational aid in a manner non-intrusive to both the user and to the individuals around the user.

Spatialized audio cues are generated using an audio filtering technique known as binaural audio filtering. With binaural audio filtering, audio signals are separated into left and right audio channels and are filtered using a Head Related Transfer Function (HRTF). Depending on how well the HRTF matches the user's head profile, binaurally filtered audio listened to through a pair of regular in-ear headphones can sound as if the audio is coming from the world around the listener as opposed to from the headphones in the listener's ears [1].

This document will first explain the objectives of this project and review similar existing projects, showing the areas not covered by current solutions. After that, this document will discuss the design requirements and specifications of the SEES Project, including the system's inputs, outputs, and processes. It will also explain the various considerations made in finalizing the design, from researching the best depth sensor for the system and determining the ideal processing methods for transforming a visual signal into a binaural audio signal, to discussing the considerations for the design of the peripheral unit and for powering the system. User walkthroughs of a simulated version of the system will then be analyzed followed by a section on the conclusions drawn from these design decisions and recommendations made for future implementations, such as in the 499 design course in Summer 2015.

2.0 Scope

As with the earlier report, the scope of this project, and by extension, the scope of this document covers topics relating to the development of an assistive navigation device for blind and partially blind individuals. These topics include computer hardware, computer

software, visual and aural signal processing, acoustics, user interface development, electrical constraints and software simulation.

One important note on the scope of this document is that 1) due to the nature of this course, the focus is primarily on design rather than on implementation or prototyping, and 2) due to the monetary constraints - namely, that the budget for this project was nonexistent - the focus of this document and of the methods and considerations outlined within favours research and simulation over prototyping and experimentation.

3.0 Review of Existing Projects

In the course of researching and defining the project, a number of existing similar solutions were examined. One particular project named ARGUS FP7 was unique in that it also uses binaural audio for navigational cues. The primary difference between this project and the ARGUS FP7 project is that the ARGUS FP7 project relies on GPS positioning and requires pre-processed environment data in order to provide navigation information to the user [2]. The intention for the SEES Project is for the device to be a self-contained unit that can gather, process, and provide feedback to the user information about the environment in real time without the need for pre-processing.

Researchers in the Computer Science and Engineering department at the University of Nevada have also conducted a number of studies and built several prototype systems for assisting the blind with navigation. There are two major hands free systems developed by these researchers, one called Headlock, and the other GIST [3, 4].

Headlock is designed around Google Glass and focuses on directing users toward predetermined object types, such as doors, by alerting the user when one has been detected and by providing audio feedback when the user veers off course [3]. GIST has users wear a depth sensing camera, and implements a gesture interface to allow users to explore the physical space in front of them [4]. Unlike Headlock, the SEES Project seeks to provide feedback on all obstacles and convey a better sense of surroundings to the user, not focusing on target-driven navigation. Similarly, while GIST focuses on object interaction and object exploration, SEES focuses on environment mapping and environment exploration.

Additionally, the feedback provided by Headlock and GIST takes a descriptive or symbolic approach. GIST will use verbal cues to inform the user of object information [4], and Headlock was tested using both verbal cues and sonification, where a steady tone indicates a course with no veering, while frequency of beeping indicates degree of veering from target [3]. In contrast, the SEES project uses a more natural, sense based approach that tags objects with audio cues that can be naturally picked up on and distinguished by the user.

4.0 Development Plan

The project was divided according to the four milestones in accordance to course schedule. A main goal was set for each of the milestones in a way that the development of the concept was as smooth as possible following a linear development lifecycle.

The first milestone began by reviewing the existing navigational aid projects and brainstorming ideas for the new system's design. Together with the analysis of the other projects, the possible main units for the product were researched such as types of sensors and the need for an auxiliary board.

For the second milestone, the project was divided in software and hardware designs. Two Software Engineering members of the team became responsible for researching the software components of the project, while the other three members began development of the hardware design of the system. During this phase, adjustments were made to the scope of the project as well as the accessibility assessment to certain components as necessitated by the results of the research.

The third milestone involved prototyping and testing the SEES system with actual users. The data acquired from testing was compiled into a report that was to be used in deciding whether the system was coherent with project expectations or not. This part of the development was the most hands-on experience, which provided the group with an approximation of the final form of the system.

The final milestone for the project covers the entire design specification for the SEES project and and an evaluation of the possibilities for its realization in the follow-up engineering course.

5.0 Project Requirements

These requirements are considered to be the minimum requirements to fulfill the basic use case of assisting a blind or visually impaired individual navigate an unfamiliar environment while keeping the design within reason.

- **System Speed:** Must be fast enough to perform 2 binaural audio transforms (left and right ear) for a given audio signal at 44000Hz in real time. (i.e. audio runs smoothly with no skipping)
- **System Memory:** Must be able to contain binaural audio software and associated control software
- **System Capability:** Must be able to output audio to headphones
- **Sensor Capability:** Must be able to sense objects in indoors, outdoors, well lit, and dimly lit environments.
- **Feedback Speed:** The time from when the sensor detects something to when the user receives feedback must be no longer than 15 milliseconds. (~60Hz)
- **Usability:** The user must find the system usable and intuitive enough that with minimal training he or she can perform basic object recognition and simple room navigation tasks in under a minute on the first tries.
- **Prototype Budget:** The system must cost under $100 to prototype and roughly $0 to design.

6.0 Project Design and Specifications

6.1 System Inputs

The SEES software takes in 3 primary inputs: A depth image stream; a Head Related Transfer Function (HRTF) profile; and a set of possible user control inputs. These inputs have been summarized in Table 1. The depth image stream is provided by the SEES peripheral. Using the points on the depth image, the software builds a model of objects and obstacles relative to the user's head. The model is then used to choose transfer functions from the HRTF profile stored on the host processing device. Finally, the software computes spatialized audio cues based on the object model using the chosen transfer functions. In addition to the sensor input and HRTF profile, the software also takes in inputs from the user to allow actions such as switching operating modes or loading and changing HRTF profiles.

Table 1: System Inputs

	Description
Image Stream	A depth image stream mapping points on the image to distances.
HRTF Profile	A Head Related Transfer Function profile containing transfer functions necessary for computing binaural audio cues.
User Input	Control inputs from the user to change operating modes or configure the software.

6.2 System Outputs

With the exception of a user interface that may include graphical, aural, and haptic feedback, the SEES software's primary output is one or more binaural audio cues transmitted to the user as a single stereo audio stream through a pair of headphones. In order to capitalize on the effective range of the human auditory system, audio cues are sampled at 44000Hz. This allows the production of audio signals of up to 20000Hz which is the maximum range for human hearing [5].

Table 2: System Outputs

	Description
Audio Stream	An audio stream containing binaural audio cues streamed at 44000Hz
Menu Feedback	System response to user input in the form of graphical, aural, and/or haptic feedback.

6.3 System Audio Processing

The SEES Software uses binaural audio filtering to generate spatialized audio cues when listened to through a pair of headphones. The process involves taking a single mono audio cue used as a base and filtering it with two transfer functions for the left and right ears to obtain two audio signals that are then streamed out to headphones as stereo audio. Selection of the two transfer functions to use in filtering the audio is dependent on the desired perceived location of the spatialized audio cue. Typical HRTF profiles contain transfer functions for each of the subject's ears recorded at various azimuths and elevations around the subject's head. Thus, given a pair of horizontal coordinates in which an spatialized audio cue is desired, the corresponding transfer function can be chosen appropriately. Figure 1 summarizes the process of generating binaural audio by showing how the process can be implemented in a system known as a convolvotron:

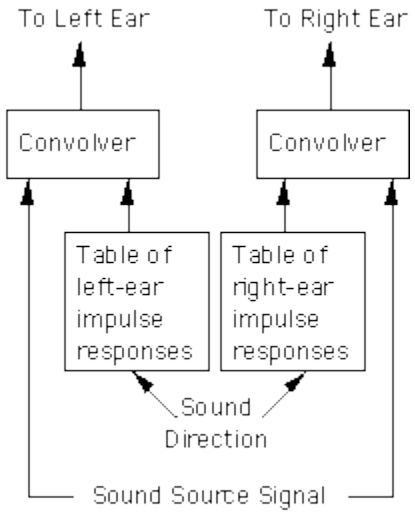


Figure 1: A binaural audio convolvotron as illustrated by the University of Calgary CIPIC Interface Laboratory HRTF Database [16]

6.4 System Interface

The SEES software has 3 primary operating modes outlined in Table 3. The software can be modeled as a state machine, as shown in Figure 2. The entry point of the software when launched is in the idle state, where the system is not actively interfacing with any of the inputs or outputs. The user can transition into the sensing or the configuration modes from here.

Note that in the following paragraphs term 'selection' may imply pressing a button on the graphical interface or prompting the system using a voice command.

Configuration mode is entered from idle mode by the user selecting the *Configure* option. The primary feature of configuration mode is to allow the user to set up his or her HRTF profile. This can be accomplished in one of two ways:

- Download the profile from a database of pre-recorded HRTF profiles.
- Import a custom profile from storage on the mobile device

Configuration mode can be exited by the user selecting the *Back* option. Any changes will be saved. The user will be returned to idle mode.

Sensing mode is entered from idle mode by the user selecting the *Begin Sensing* option. When in sensing mode, the system will actively interface with the input and output. The system will poll the sensor input and perform convolutions in real time to transform the input into spatialized audio cues. The system will send this to the output. Sensing mode can be exited by the user selecting the *Stop Sensing* option, which will return the software to idle mode.

Table 3: Main Running States

State	Description
Sensing	System will interface with sensors and headphones, performing convolutions to transform the visual signal to an auditory signal based on the user's HRTF profile. System will monitor user input to switch to idle.
Idle	System will monitor user input to switch to sensing mode or to enter configuration.
Configuration	System will allow user to set up operating modes - specifically, user can download, import an HRTF profile.

Table 4: State Transitions

Transition	Description
Sensing → Idle	System will transition from **Sensing** to **Idle** when user selects the *Stop Sensing* option in the application.
Idle → Sensing	System will transition from **Idle** to **Sensing** when user selects the *Begin Sensing* option in the application.
Idle → Configuration	System will transition from **Idle** to **Configuration** when user selects the *Configure* option in the application.
Configuration → Idle	System will transition from **Configuration** to **Idle** when user selects the *Back* option in the application.

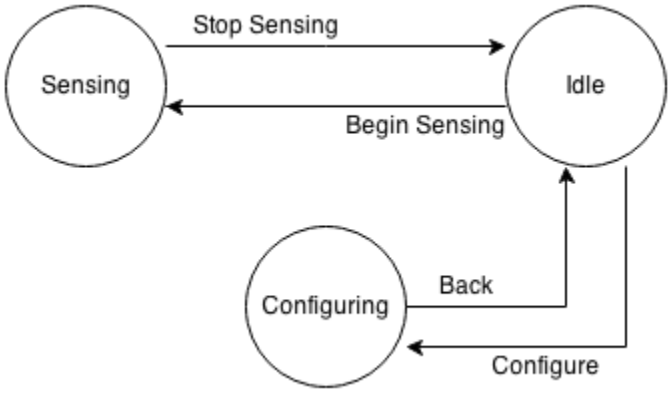


Figure 2: State Machine diagram for SEES software operating modes

7.0 Considerations

This section will describe the hardware and software considerations for the SEES Project. The main hardware component is the SEES peripheral - a device worn by the user that holds the sensor or sensors used to acquire depth information about the user's environment. The information gathered by the sensors is sent to the system software on the host device which then provides appropriate feedback to the user.

The main considerations for this include the design of the peripheral device itself, the type of sensor or sensors used in the peripheral, the nature and function of the signal processing unit, including the format specification of the data that is captured, and the minimum operating electrical requirements of the system as a whole.

7.1 Peripheral Design

One primary consideration of the system was the design of the device's sensing peripheral and how the sensing device should be used by the user. Some possibilities discussed were mounting the sensors on a cane or wand held in the user's hands, on a vest worn on the user's torso, or as part of a headset worn on the user's head.

For the wand type system, the user uses the device by pointing it in the direction they wish to detect. This allows the user to freely sweep the wand across the scene and listen for the resulting audio cues. Because the device moves independently of the user's head, the user is free to adjust the orientation of their head to better understand the distance and orientation of binaural audio cues. However, this introduces the need to reconcile both the device's position and orientation with the user's head in order to properly align the generated binaural audio cues with the world around the user. Additionally, having the user constantly sweeping the scene may introduce fatigue in the user's arms over long term usage of the device.

For the vest type design, the sensor is limited detecting in a field in front of the user. Binaural audio cues are used to signal where in this field objects and obstacles are located. Similar to the wand type design, the user may also freely tilt or rotate their head to clarify the nature of the device's audio cues. However, unlike the handheld, the relative position of the sensors to the user's head can be assumed to be constant. As a result, only orientation needs to be reconciled between the user's head and the device. One drawback of this particular design is that the device may not always be pointed in the direction that the user wishes to observe.

For the final design: the head mounted design, sensors are mounted to a headset that is worn on the user's head. This sensor layout would be the most natural choice for replicating the natural human field of view. As the device is mounted to the user's head, the user is free to direct the device as they would their vision. Additionally, because the device is always positioned and aligned to the user's head, no complex calculations are required in aligning the binaural audio cues to the world around the user's head. That being said, using a head mounted device also means that the user is limited to how much they can adjust their listening angle before the target audio cue drops out of the device's field of vision.

Of these designs, the head mounted design was chosen to be used for preliminary prototype designs. The head mounted device avoids the need for difficult orientation and position calculations and can be used most similarly to regular human sight. While the design does limit the sound localization capabilities of the user, it does not eliminate them completely. Figure 3 shows an early mockup of how this particular design could be constructed similar to a pair of spectacles.

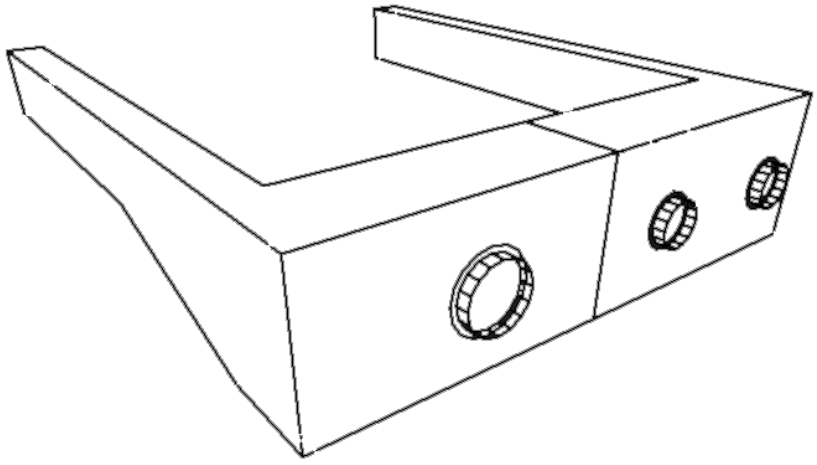


Figure 3: Prototype Design Mockup of Headset Type Peripheral

7.2 Sensor Type

The first and perhaps the most difficult question to answer was: Which sensor or sensors should be used? The sensor is one of the most important aspects of this system - with an inadequate sensing method, the entire system is effectively useless. It was therefore imperative to find the fine balance between usability and feasibility, and between functionality and cost.

7.2.1 Infrared

Infrared, or IR, is one of the most readily available options to be considered. [6] examines two common types of IR sensors which exist:

1) Sensors which provide binary output
2) Sensors which provide analog output

The sensors with a binary output are used for detecting object proximity, but not range. We will not be considering these, as depth is an absolute requirement for this system. The other IR sensors are range-finding sensors, which supply measurements of the actual distance of a point on an obstacle from the sensor. This is also known as threshold distance.

Simple infrared range-finding sensors are both cheap and effective at gathering depth information at a single point, but they are not very well suited for mapping a complete environment all at once. Certain conditions prove to be less than ideal for infrared sensors, as bright sunlight can interfere with depth mapping due to saturation of the sensors. Similarly, dark and low contrast objects can be difficult to accurately identify.

For longer range infrared sensors, the minimum range also tends to be quite large [7]. For this reason, multiple sensors of varying ranging would be required with infrared.

7.2.2 Ultrasonic

Ultrasonic sensors are based on sound rather than light for ranging, and are therefore not affected by ambient lighting conditions. Ultrasonic sensors can be used to measure the distance to an object by emitting an ultrasonic pulse. The sensor then measures the propagation time for the echo to return back to the sensor as well as the resulting frequency. These values are then used to calculate the distance to the target. It is important to note that ultrasonic sensors are impacted by sound absorbing materials such as sponge like surfaces and by ghost echo - reverberation providing additional delayed feedback to the sensor, which can produce inaccurate results. This has the potential the limit the frequency of accurate measurements obtainable by the sensor. Despite being easily handled, these sensors do not supply further spatial information from the object.

In order to attain a more complete depth range analysis, multiple sensors may need to be used. More complex timing considerations would then need to be made in order to avoid interference from the bouncing echos of the different sensors. Ultrasonic sensors also have proximity ‘dead zones’, where close objects may be outside of detectable range [8].

Ultrasonic sensors also tend to be more expensive than infrared sensors, an important consideration in the nature of a project such as this, where the budget for prototype testing is minimal.

7.2.3 Time of Flight Camera

Another approach is to create a wide range depth map utilizing a time of flight camera. Time of flight (ToF) cameras operate by using infrared light to illuminate the scene and then measure the time of flight for the light to return [9]. This makes it possible to determine the depth of an entire image as opposed to a single point, eliminating issues present with simple infrared range-finding and with traditional ultrasonic sensing.

Because ToF cameras typically use infrared light, this method will suffer from some of the same limitations under bright light as simple infrared range-finding does. [10] discusses three different ToF cameras and how well each performs object recognition under bright light (specifically, identifying a leaf) with varied success.

Using a ToF camera would mean the system could operate on a single sensor, as the depth map provides a large view of the immediate area, similar to the cone of an ultrasonic sensor, but without the concern of ghost echoes.

A significant limiting factor of ToF cameras is the cost. Many of the popular ToF cameras cost around $10,000 [11].

One existing ToF system that has become popular in research is the Microsoft Kinect. A search of peer reviewed journal articles for ‘Kinect’ in the UVic summons system shows over 1300 articles. The Kinect provides a relatively cheap and high resolution (640x480) ToF camera running at 30 FPS. 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. The limitations of the Kinect are dull or shiny surfaces viewed at sharp angles, invalid measurements of areas not illuminated by the infrared beam, and bright daylight [9]. Bright daylight does cause saturation of the depth image, leading to potential outliers or gaps.

Because of the popularity of the Kinect, there has been research into improving the results of its data maps, which contain the distance, in millimeters, to the nearest object at the corresponding (x,y) coordinate in the depth sensor’s field of view. Of particular interest are [12] and [13], which discuss novel approaches to replacing missing values based on estimations from nearby valid values.

It should be noted that the Kinect unit itself has multiple peripheral components in addition to its ToF sensor including a tilt motor, an array of microphones, and a color image CMOS sensor (Figure 4). As these components are not necessary for sensing depth, only the ToF sensor and the corresponding processor components would be integrated into the SEES peripheral.

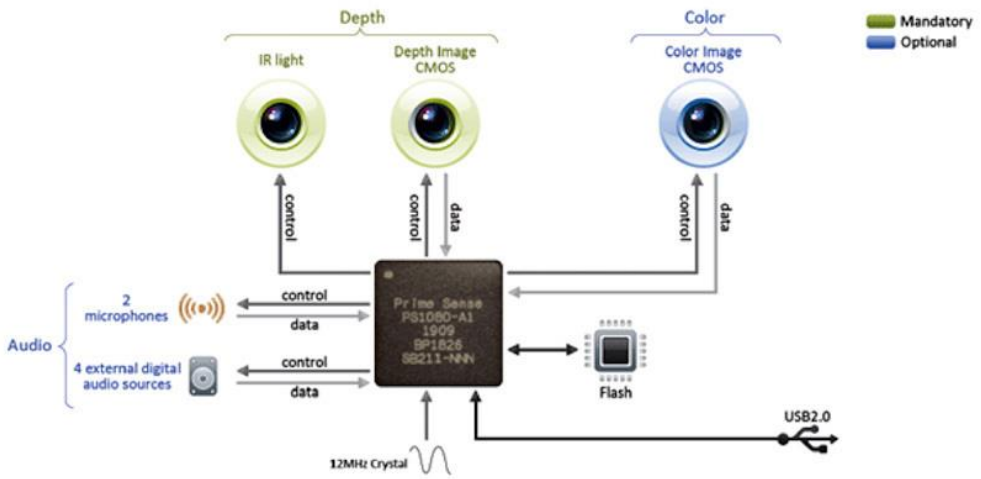


Figure 4: Block diagram of the Kinect sensing unit designed by PrimeSense [14]

Table 5: Sensor Comparison

State	Range Max	Range Min	Limitations	Cost
Infrared	5m	5-100cm	Bright light; interference	$15
Ultrasonic	10m	25cm	Sound absorbing materials; ghost echo	$55
Time of Flight	5m - 5km [15]	1cm - 50cm	Bright light	Wide variation (many ~$10k)
Kinect (ToF)	6m	80cm	Bright light	$150 ($50 used)

7.2.4 Sensor Conclusion

Table 5 below shows a comparison of basic infrared range-finding sensors, ultrasonic sensors, general time of flight sensors, and the Kinect.

The limitations of simple infrared range-finding sensors outweigh the simplicity and the cost. With poor distance ranges leading to the need for multiple sensors and therefore the need for additional calibration to avoid interference, and the poor performance of these sensors in bright light, these sensors would likely not fulfill the project's requirements.

Ultrasonic sensors reasonably fulfill the sensing requirements, but have a much higher cost than infrared. If more than one sensor is required, the price will quickly increase beyond a reasonable budget. Also, for indoor solutions, the ghost echo problem would negatively impact the validity of the sensor data.

The general industry level ToF cameras can obviously not be considered here due to the prohibitive price alone. The Kinect, however, provides a reasonable alternative, offering a self-contained sensor array with a depth map larger than a single point and without ghost echoes. Because the Kinect does have limitations in bright light, once hardware testing begins (which was outside of the scope and budget for this design project),

The precise spatial impediments experience in such conditions will determine if a single ultrasonic sensor should be used in conjunction with the Kinect.

7.3 Processing Technique

Regardless of what type of sensor is used, a processing unit is necessary to run a HRTF and generate the binaural audio from the sensor data. Preferably, the processor should be able to run C/C++ code and be fast enough to perform 2 binaural audio transforms (left and right ear) for a given audio signal at 44,000Hz in real time (i.e. audio runs smoothly with no skipping). The processing unit under consideration is a smartphone device running the Android Operating system. Smartphones meet the required specifications, in processing power, and also by being an accessible device already owned by many individuals.

As traditional convolution is CPU intensive to calculate, convolution for the left and right audio signals will be performed using fourier analysis via the convolution theorem. Using the convolution theorem, convolution between two signals can instead be calculated as a point-wise multiplication of the two signal's fourier representations [16]. This changes the number of point multiplications required in the calculation from n^2 down to n; however, it also introduces the need for efficiently calculating the fourier transform of the streaming audio signal.

Assuming that the HRTF fourier representations can be pre-calculated, then producing spatialized audio requires 1 forward transform for the mono input channel and 2 backward transforms for the stereo output channels. Therefore, in order to produce continuous filtered audio for a signal provided at 44,000Hz it is necessary to calculate fourier transforms of 3 x 44,000 = 132,000 data points per second.

To this end, the software will use the native C FFTW fourier transform subroutine library, which is one of the fastest and most used FFT libraries. The FFTW library has been benchmarked as being able to perform nearly 4000 4096-point fourier transforms per second on a 1.06 GHz processor core [17]. This can equate to roughly 16,000,000 point calculations per second on a mobile phone processor of the same strength. This makes the FFTW library an ideal choice for implementing a continuous convolution on a mobile platform. In android devices in particular, FFTW has shown to be faster than Java FFT and to have a better performance when it is used with single-thread rather than multiple-threads,

which is an important characteristic from the point of view of the device's power consumption [18].

7.4 Power Constraints

7.4.1 Sensor

When using Microsoft Kinect, the device can be separately acquired as Kinect for Windows as well as Kinect for Xbox. These two versions have a power supply cable attached which provides the voltage for the device. Differently, the newer version of the Kinect has a special USB port which is capable of delivering enough energy (12V) to power the device. The problem is that standard USB computer ports deliver only within 5% plus/minus 5V and 2.5W. Recently, Microsoft has released a new cable, Kinect for Windows, which allows users to connect their Kinects to computer in a mix between special USB connection going into the kinect, which forks in a common USB plus a power input, as shown in the figure 5 below.

Figure 5: Microsoft Kinect connection guide the Xbox 360

The case illustrated above can be a solution but it still limits the scope of our project due to being limited in distance by the length of the supply cable. The Kinect for Windows has a hub which allows the cross between USB 3.0 and the power supply, possibly transforming the voltage inside and stepping it down to 12V. However, it is unfeasible to say what exactly happens inside the hub, since Microsoft did not open the hub architecture for general public, then limiting the specs and changes that can be done in order to adapt it to diverse usages. As a result, the hub expects relatively high voltages, it becomes almost impractical to use it for powering the device.

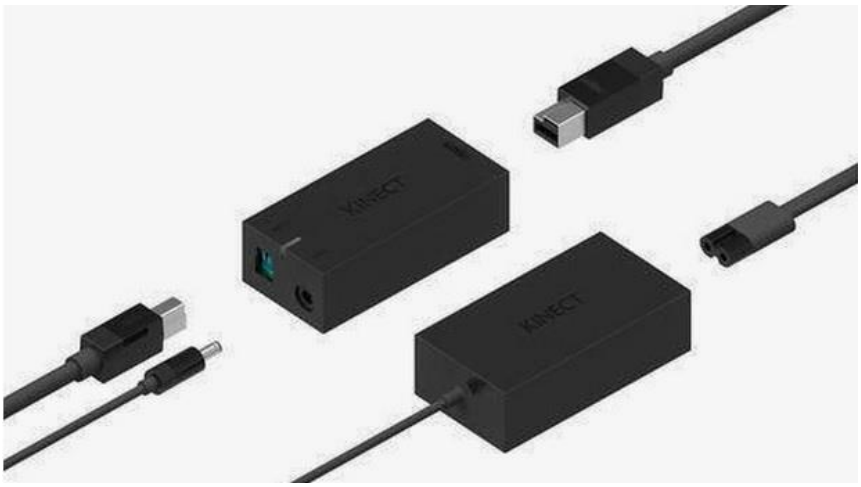


Figure 6: Microsoft Kinect power adapters for Xbox360 (left) and PC (right)

Another possibility to be considered is the Kinect for Xbox 360 console, which has a different type of communication/energy supply cable. This cable does not count on a special hub, and expects a regulate 12V supply. Online projects as ROS.org's have come up with solutions for powering the kinect with a circuit regulator in 12V [19]. This opens up some possibilities for the project, since the regulator can take higher voltages and filter them with satisfactory accuracy. Parallelly, the market offers different options for 12V batteries according to the need, from a single unit to many rechargeable AA's in series. Then, the prime requirement for the battery is to be light and easily rechargeable, since simple regulators could adapt the voltage to 12V and current to 1A as needed.

What seems to be the most suitable option for our project is stripping down the Kinect to extract the depth camera, IR emitter, and processing chip, developed by a company called PrimeSense. This component works by using its IR dots and capturing simple CMOS images with an IR filter. The strongest advantage of using this stripped down is the absence of servos, allowing the device to work powered only with the standard 5V USB port, which seems to be enough to power the sensors and camera.

7.4.2 Mobile Power Autonomy

Mobile power autonomy is also a concern that has to be taken into consideration when running large Fourier Transforms. Researchers from the University of Waikato benchmarked the number of MFLOPS (Million floating point operations per second) for different types of Fourier Transforms [20]. As a result, they obtained that a 131072-point FFTS (The Fastest Fourier Transform of the South) running on a Cortex A9, processor unit contained in the Samsung Galaxy SIII, takes approximately 1200 MFLOPS. If comparing the amount of MFLOPS used to run the transform with the capacity of the processor, which is 1.5 times 4 GFLOPS due to the number of cores in the mobile, it is possible to conclude that even with a large number of points, the fourier transform would take less than a fourth of the overall capacity offered for these operations. As a result, the processor units are able to perform these operations without overheating, which would lead to a battery drain. It is also important to consider the user awareness to avoid execution of applications that also need heavy processing loads from the device.

8.0 System Testing

8.1 Test Setup

In order to evaluate usability of the design decisions stated above, testing was done using a software prototype to approximate the characteristics of the chosen hardware. The base setup for the prototype consisted of a controllable user avatar placed inside a virtual environment. Audio sources could be placed inside the virtual environment which would generate binaural audio cues according to the position and orientation of the virtual avatar. These cues were output to the user through a basic pair of stereo in-ear headphones. Finally, a head tracking device worn by the user allowed for binaural audio cues to be readjusted as the user moved their head.

For simulating the SEES headset style peripheral, a virtual depth camera was positioned at the avatar's head and oriented to the user's head orientation. The depth camera was configured with a maximum depth range of 50 virtual units corresponding to 3.5m in real world units. For this test, cue positioning used a 16-point mapping layout which mapped audio cues to 16 equidistant points across the centre horizontal of the depth image field. Additionally, to further assist in distinguishing each audio cue, each cue consisted of a uniquely pitched tone starting with the highest of 523Hz (C_5) being at the centre of the user's vision and then scaling down to 104Hz ($A^{\flat}_2$) at the peripherals of the user's vision. Figures 7 and 8 illustrate the depth image and audio cue mapping for this setup. Despite having 640x480 depth points available from the depth camera, this minimalistic layout was chosen both to minimize the amount of confusion for test participants using the new system as well as evaluate the feasibility of using lower resolution depth sensors for the headset. That being said, more testing will be required in order to determine the most understandable and useful mapping layout.

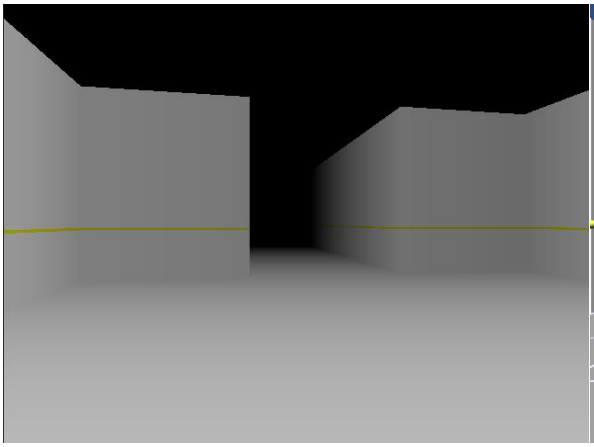


Figure 7a: A depth image generated using a virtual depth camera. The centre horizontal has been highlighted.

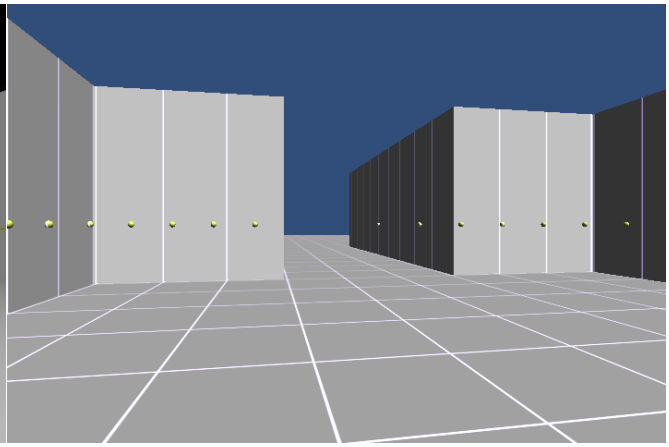


Figure 7b: The corresponding scene with the 16 spatialized audio cues mapped into it. The 2 centre cues are at maximum distance and are neither shown or heard.

8.2 Test Protocol

Before testing, test participants were first given an explanation of the SEES project and its goals followed by an introduction to the test prototype and instructions on how to control the virtual avatar. Participants were then asked to put on a pair of headphones, the head tracking device, and a blindfold to begin testing. While most participants found the default binaural audio profile sufficient, participants were also given option change profiles to find one that comfortably matched their own.

The first test tasked participants with navigating their virtual avatar towards a randomly placed audio signal in the scene around them. This test did not use the SEES audio cues, but instead focused on measuring the participant's ability to echolocate generated binaural audio cues as well as testing the participant's comfort with navigating in the virtual environment. This test recorded the amount of time taken by the participant in completing the task as well as the level of confidence in which they were able to navigate.

Following the first test, the SEES audio cues were enabled and participants were given a short session to get accustomed to the audio cue based navigation system. Using this system, participants were moved to the next test which focused on shape recognition. Participants were presented with a basic shape at a distance of 1 metre from their virtual avatar and asked to try and identify the shape using the system. The test was then repeated for the 3 other shapes for a total of 4 tests with 4 different shapes. Figure 9 shows the 4 shapes that were used for testing: a rectangular speed limit sign, a triangular yield sign, a diamond shaped turn sign, and a octagonal stop sign. This test recorded each shape and the participant's corresponding guess as well as the time taken before making the guess.

The 3rd test focused on the system's ability to allow participants to distinguish and count individual objects placed in a scene in front of them. The number of objects ranged from 2 to 5 objects and were placed no closer than 1m and no further than 2.5m from the participant's avatar. Participants were encouraged to voice their thoughts as they tried to distinguish and count each of the objects in the scene. This test recorded the number of shapes presented to the participant, their corresponding guess and the time taken to make their guess. It also recorded the number of times a participant counted the same object twice as well as the number of objects that the participant overlooked.

The final test tested the usability of the system for navigational purposes. The participant's virtual avatar was placed with a random position and orientation inside a square, 2.5m^2 room with a single 1m wide exit located at the middle of one of the room's walls. Participants were given an explanation of the nature of this room and were asked navigate out the room's exit. This test recorded the amount of time taken by the participant in completing the task as well as the level of confidence in which they were able to navigate.

Figure 8: The 4 shapes used for testing.

8.3 Test Subjects

Test subjects were primarily limited to university students with no visual or auditory impairments. While this introduces some bias with regards to usability for the target user base of visually impaired individuals, the data acquired from these test subjects is applicable for users of the system who may have little or no experience navigating without sight. That being said, future tests - particularly tests including actual SEES hardware - should include both visually healthy and visually impaired individuals.

8.4 Results and Observations

Table 6: Test 1 task completion time per participant

Participant #	Time (s)
1	30
2	40
3	150
4	50
Average:	***67.5***

For the first test, participants were able to locate the direction of the target in relation to their virtual avatar almost immediately. That being said, the majority of the participants were apprehensive when it came to navigating the virtual environment leading to a fairly large average task completion time of 67.5 seconds. This is an understandable outcome as navigation in the virtual environment deprived the participants of many of the common cues and sensations that are experienced in real world navigation. Additionally, the lack of real-world reverberations of the simulated audio cues made it difficult for some participants to determine the magnitude of the audio cue being played. As a result, these participants were unable to determine their distance from the cue and how far they needed to move to reach it. This implies that reverberation is something that will need to be researched further and integrated into the SEES device.

Table 7: Test 2 number of correct identifications and average identification time per shape

Shape	Correct Guesses (of 4 participants)	Average Identification Time (s)
Speed Sign (Rectangle)	4	28.7
Yield Sign (Flipped Triangle)	3	48.7
Turn Sign (Diamond)	3	27.5
Stop Sign (Octagon)	1	65.0

For the second test, participants typically had the least difficulty in identifying the rectangular speed sign while the most difficulty was found in identifying the octagonal stop sign. For the stop sign, the most common outcome was the participant identifying it as a circle. The minimal 16-point audio cue layout and the operating distance of 1m meant that only a maximum of 7 of these audio points in the participant's field of view could be positioned on the stop sign at any given time. Therefore, the confusion between an octagon and a circle can be best attributed to the low resolution of identification points provided by the prototype system. This means that sensor resolution as well as the corresponding number of depth points fed back to the user is important for distinguishing between shapes with smooth and angular edges.

One final observation about the second test was that one participant was initially confused by the audio cue layout such that shapes sounded inverted to them. That is, the triangular yield sign sounded thinnest at the top and widest at the bottom while the octagonal stop sign sounded widest at the edges and thinnest at the centre. After having a bit more practice with the system, the participant was able to successfully identify the shapes. However, this indicates that some audio cue layouts may seem intuitive for some individuals, but not for others. As mentioned previously, more testing will be necessary to determine the most comprehensible layout.

Table 8: Test 3 task completion time and number of miscounts per participant

Participant #	Completion Time (s)	Objects Doublecounted	Objects Missed
1	180	1	1
2	150	1	1
3	150	0	0
4	60	0	2
Average:	***135***		

The third test took longer overall than the other tests with completion of the task averaging at 135s. Participants often made good use of the distinction between peripheral audio cues and focal audio cues when distinguishing between objects that were far apart. In contrast, the main issue encountered by participants when counting the objects was distinguishing

between objects that were in close proximity to each other. This is likely caused due to difficulties in hearing the silence or 'auditory gap' between audio cues resting on one object and audio cues resting on the other. This implies that instead of using silence to represent the non-presence of an object, rather a distinct audio cue should be used as a negative to assert explicitly when a void rests between two sets of audio cues.

Table 9: Test 4 task completion time per participant

Participant #	Time (s)
1	70
2	45
3	100
4	40
Average:	***63.7***

The final test, each participant was able to successfully exit the room. Some were faster than others, but the average completion time was 63.7 seconds. Each participant was seen taking deliberate actions to determine the direction of the exit as opposed to randomly trying to find the exit. For example, most of the participants started the test by first scanning the room in 360 degrees using the system. While this usually worked, one participant had their head held at a downward angle causing the audio cues to be emitted from the floor as opposed to the walls. Upon locating the exitway, participants were able to identify the centre part of the opening by listening for the combination of audio cues that had audio on their peripheral view with silence in their focal view. Following this, they were able to successfully exit the room and complete the task.

One final observation to be made with the overall testing is that the data recorded for these tests were of first time users of the system. Repeat testing with one participant showed that task performance and completion time could be significantly improved as the user became more accustomed to the system. This implies that the SEES device may need some degree of training before it can be effectively utilized by a user or blind individual.

9.0 Conclusion

In this project, an initial design was drafted for the purpose of assisting the visually impaired with navigating the world around them using a natural, sense based approach.
Several brief but important requirements were set out, covering system speed, memory, and capability, sensor capability, feedback speed, usability, and budget and it was shown how existing solutions do not fit the same space this project aims to fill.
Specifications were drafted, focusing on system input, output, and process. For input, the specifications of an input image stream, HRTF profile and user application were outlined, while in the system output section the resulting binaural audio stream output and menu feedback were specified. Finally, the specifications for audio and user interface processes were detailed.

Taken in conjunction with the requirements and specifications, the considerations and limitations resulted in a number of successes for the SEES design, as well as some limitations. These will be discussed below.

9.1 Successes

The sensor considerations concluded that the Microsoft Kinect, a cheap ToF camera, would be the best alternative for sensor choice. This sensor is ideal in that it meets the needs of SEES by falling easily within budget when purchased used, captures a full depth map picture which can be used to easily extract depth points demonstrated in the simulations, can sense in most lighting conditions, and does not suffer from ghost echoes. It was also found that the entire bulk of the Kinect will not prove necessary to include in the device.

Research into the processing technique revealed that the intensive binaural audio convolution necessary to be performed can run on a regular modern smart phone. Therefore, additional audio processing hardware will not be necessary for the system.

An examination of a variety of peripheral device setups concluded that a head mounted design, where sensors are mounted to a headset worn by the user, provides the greatest freedom and least distraction and does not require the user to learn any new movements or motions to use.

For power considerations, PrimeSense's device appears to be a feasible option when powered by the USB connection. The Kinect as a whole, is not an option to be dismissed, but considered as a contingency plan. Even though Microsoft's device depends on 12V, this voltage can be easily achieved with simple combination of small-sized batteries.

Finally, the user testing was critical in validating the SEES binaural audio cues and high level system concept. While limitations were present in regards to the test subjects chosen and the nature of the virtual test environment, Positive performance results were obtained with first time users, who were not familiarized with system and all participants were able to complete successfully or mostly successfully the tasks within a reasonable time. As a result of this, it can be concluded that the system shows a potential to work effectively as intended when realized.

9.2 Potential Limitations

In spite of the encouraging successes in this project, there are a number of design limitations and potential design limitations which must be considered.

The sensor system is known to suffer in direct sunlight, though few sources could be found that examined this in depth. The maximum range of the sensor could potentially limit the practical usability for a blind individual. Finally for the sensor system, the parts of the Kinect needed are only a subset of the full device, and so overhead exists in needing to strip it down.
Binaural audio processing may be intensive enough to interfere with other user mobile applications, depending on the intensity of those applications.

The interference of the system's audio with the user's normal hearing may be too destructive for the system to be practical.

Powering the Kinect may be complicated, as it must first be stripped down. Without stripping the Kinect down, a bulky energy source would be necessary and the user's mobile phone may not be able to drive the entire system on its own.

Finally, the user simulated testing, while valuable, does not necessarily show the same result that a physical test with a visually impaired individual would.

10.0 Recommendations

Based on the research and user walkthroughs conducted for this project, it is the recommendation of this report to develop a headset style peripheral device with a time of flight depth camera built in and used to capture proximity and distance information from a user`s environment. It is recommended to strip down the Kinect and use only the sensor component within. It is recommended to connect the headset peripheral to a user`s mobile phone and use the phone to process the incoming visual signal into binaural audio using native C fast Fourier transforms.

It is the recommendation of this report to first and foremost develop an extensive system testing plan for 499. There are 5 things identified as needing further testing as hardware prototypes.

1. The basic Kinect sensor array must be tested in a variety of settings to determine fully the distance and lighting conditions under which it can work.
2. CPU processing of convolutions must be tested on a variety of mobile devices to determine the minimum operating requirements for this system.
3. Power consumption must be tested (maybe how long an average phone could drive the entire system for; that or what sort of external battery pack the user needs)
4. Test walkthroughs of a system prototype must be conducted with many users, including visually impaired individuals. Tests must be conducted early and often, and updates based off user feedback.
5. The reduction of sound localization for the user while using this system must be tested.

References

[1] R. H. Y. So et al., “Effects of spectral manipulation on non individualized head-related transfer functions (HRTFs),” Human Factors, pp. 271-283, June 2011.

[2] E. Carrasco et al., “Autonomous navigation based on binaural guidance for people with visual impairment,” Assistive technology: From research to practice, assistive technology research series, pp. 690-694, Sept. 2014.

[3] A. Fiannaca, I. Apostolopoulous, and E. Folmer, “Headlock: a wearable navigation aid that helps blind cane users traverse large open spaces,” in Intl. ACM SIGACCESS Conf. ASSETS’14, 2014.

[4] V. Khambadkar and E. Folmer, “GIST: a gesture interface for remote spatial perception,” Proc. ACM Symp. UIST’13, pp. 397-404, Oct. 2013.

[5] H. E. Heffner and R. S. Heffner, “Hearing ranges of laboratory animals,” in Journal of the American Association for Laboratory Animal Science. American Association for Laboratory Animal Science, 2007, vol. 46, num. 1, 2007, pp. 20-22.

[6] “Infrared vs ultrasonic - What you should know,” *Society of Robots,* [online] Jan. 2008, Available: http://www.societyofrobots.com/member_tutorials/node/71 (Accessed: 1 December, 2014).

[7] “Sharp infrared IR ranger comparison,” *Acroname: Automation Engineering*, [online] 2014, Available: http://www.acroname.com/articles/sharp.html (Accessed: 1 December, 2014).

[8] T. Mohammad, “Using ultrasonic and infrared sensors for distance measurement,” World Academy of Science, Engineering and Technology, vol. 51, Mar. 2009.

[9] B. Langmann, K. Hartmann, and O. Loffeld, “Depth camera technology comparison and performance evaluation,” in proc. ICPRAM (2), 2012, pp. 438-444.

[10] W. Kazmi et al., “Indoor and outdoor depth imaging of leaves with time-of-flight and stereo vision sensors: Analysis and comparison,” in proc. ISPRS, Feb. 2014, pp. 128-146.

[11] T. Deyle, “Low-cost depth cameras to emerge in 2010?,” *Hizook* | *Robotic News for Academics & Professionals*, [online] Mar. 2010, Available: http://www.hizook.com/blog/2010/03/28/low-cost-depth-cameras-aka-ranging-cameras-or-rgb-d-cameras-emerge-2010 (Accessed: 1 December, 2014).

[12] K. R. Vijayanagar, M. Loghman, and J. Kim, “Refinement of depth maps generated by low-cost depth sensors,” ISOCC, 2012 Int., Nov. 2012, pp. 355-358.

[13] M. Stommel, M. Beetz, and X. Weiliang, “Inpainting of missing values in the kinect sensor’s depth maps based on background estimates,” in Sensors Journal, IEEE, vol. 14, iss. 4, Nov. 2013, pp. 1107-1116.

[14] “Microsoft Kinect teardown,” *iFixit: The free repair manual,* [online] Nov. 2010, Available: https://www.ifixit.com/Teardown/Microsoft+Kinect+Teardown/4066 (Accessed: 1 December, 2014).

[15] D. Wison, “Time-of-flight camera has long range,” *Vision Systems Design - Machine Vision Systems and Image Processing Applications*, [online] Apr. 2013, Available: http://www.vision-systems.com/articles/2013/04/time-of-flight-camera-has-long-range.html (Accessed: 1 December, 2014).

[16] Department of Electrical and Computer Engineering, University of California Davis., *HRTF-Based Systems,* [online] Feb. 2011, Available: http://interface.idav.ucdavis.edu/sound/tutorial/hrtfsys.html (Accessed: 1 December, 2014).

[17] M. Frigo and S. G. Johnson, “1.06 GHz PowerPC 7447A, gcc-3.4, ” *FFTW*, [online] Mar. 2014, Available: http://www.fftw.org/speed/G4-1.06GHz-gcc3.4/ (Accessed: 1 December, 2014).

[18] A.D. Carvalho Jr et al., “FFT benchmark on android devices: Java versus JNI,” Comp. Sci. Dept. Univ. of São Paulo, Brasilia, Brazil, 2013.

[19] M. Wise, “Adding a kinect to an iRobot create,” *ROS Wiki*, [online] May 2011, Available: http://wiki.ros.org/kinect/Tutorials/Adding%20a%20Kinect%20to%20an%20iRobot%20Create (Accessed: 1 December, 2014).

[20] A. M. Blake, “Dynamically generating FFT code on mobile devices,” in Journal of Signal Processing Systems, vol. 76, iss. 3, Sept. 2014, pp. 275-281.

Appendix A - Work Logs

Daniel Faulkner		
Date	Time (Hrs)	Description of Task
09/09/14	0.5	Group Formation/Meeting
09/24/14	1.0	Group Selection & Clarification Emails
10/01/14	0.5	Meeting With Professor Adams
10/07/14	1.0	Project and Supervisor Selection Emails
10/09/14, 10/14/14	2.0	Group Meeting
10/10/14	2.0	Paragraph “How you envision the project”
10/14/14	0.5	Meeting With Professor Adams
10/19/14, 10/20/14	3.5	Progress Report
10/21/14, 10/31/14, 11/03/14	3.0	Milestone Planning Meeting
11/03/14	4.5	Software Design Document
11/14/14	1.5	Elevator Pitch Preparation
11/16/14	2.0	Simulation Testing
11/17/14	1.0	Simulation Testing
11/18/14	1.0	Elevator Pitch Preparation
11/29/14	4.0	User Testing; Final Report
11/30/14	5.0	Final Report (Research and testing)
12/01/14	9.0	Final Report (Research and conclusions)
12/02/14	2.5	Final Report

Paulo Tabarro		
Date	Time (Hrs)	Description of Task
09/09/14	0.5	Group Formation/Meeting
10/09/14, 10/14/14	2.0	Group Meeting
10/13/14	2.0	Paragraph “How you envision the project”
10/14/14	0.5	Meeting With Professor Adams
10/19/14, 10/20/14	4.0	Progress Report
10/21/14, 11/03/14	2.0	Milestone Planning Meeting
11/04/14	2.0	Hardware Design Document
11/14/14	1.5	Elevator Pitch Preparation
11/16/14	2.0	Simulation Testing
11/17/14	1.0	Simulation Testing
11/18/14	1.0	Elevator Pitch Preparation
11/29/14	2.0	Final Report
11/30/14	3.0	Final Report

12/01/14	6.0	Final Report
12/02/14	2.5	Final Report

Jason Lim		
Date	Time (Hrs)	Description of Task
09/09/14	0.5	Group Formation/Meeting
10/01/14, 10/14/14	1.0	Meeting With Professor Adams
10/09/14, 10/14/14	2.0	Group Meeting
10/10/14	1.0	Paragraph “How you envision the project”
10/19/14	3.0	Milestone 1: Prototype Design Visuals
10/19/14	2.0	Project Timeline Gantt Chart
10/19/14, 10/20/14, 10/21/14	3.0	Progress Report
10/21/14, 11/21/14, 11/03/14	3.0	Milestone Planning Meeting
11/01/14,11/04/14,11/05/14	7.0	Hardware/Software Design Documents
11/14/14	1.5	Elevator Pitch Preparation
11/16/14	2.0	Simulation Testing
11/17/14	1.0	Simulation Testing
11/18/14	1.0	Elevator Pitch Preparation
11/29/14	3.0	User Testing Milestone 4: Project Report
11/30/14	1.0	Milestone 4: Project Report
12/01/14	4.0	Milestone 4: Project Report
12/02/14	2.5	Final Report

Adalberto Outeiro		
Date	Time (Hrs)	Description of Task
09/09/14	0.5	Group Formation/Meeting
10/09/14,10/14/14	2.0	Group Meeting
10/10/14	2.0	Paragraph “How you envision the project”
10/20/14	3.0	Progress Report
10/21/14, 11/03/14	2.0	Milestone Planning Meeting
11/04/14	2.0	Hardware Design Document
11/05/14	1.0	Review of Documents
11/14/14	1.5	Elevator Pitch Preparation
11/16/14	2.0	Simulation Testing
11/17/14	1.0	Simulation Testing

11/18/14	1.0	Elevator Pitch Preparation
11/29/14	3.0	User Testing; Milestone 4: Project Report
12/01/14	3.0	Milestone 4: Project Report
12/02/14	2.5	Milestone 4: Project Report

Rajpal Chauhan		
Date	Time (Hrs)	Description of Task
09/09/14	0.5	Group Formation/Meeting
10/09/14, 10/14/14	2.0	Group Meeting
10/10/14	2.5	Paragraph “How you envision the project”
10/19/14, 10/20/14	3.5	Progress Report
10/21/14, 10/31/14, 11/03/14	3.0	Milestone Planning Meeting
11/04/14	1.5	Hardware Design Document
11/05/14	1.0	Review of Documents
11/14/14	1.5	Elevator Pitch Presentation
11/16/14	1.5	Simulation Testing
11/17/14	1.0	Simulation Testing
11/18/14	1.0	Elevator Pitch Presentation
11/29/14	2.5	IR vs Ultrasonic sensors; Milestone 4
12/01/14	3.5	Hardware vs Software convolution; Conclusion
12/02/14	2	Table of contents/Figures, Editing
12/02/14	2.5	Final Report

Appendix B - Milestones

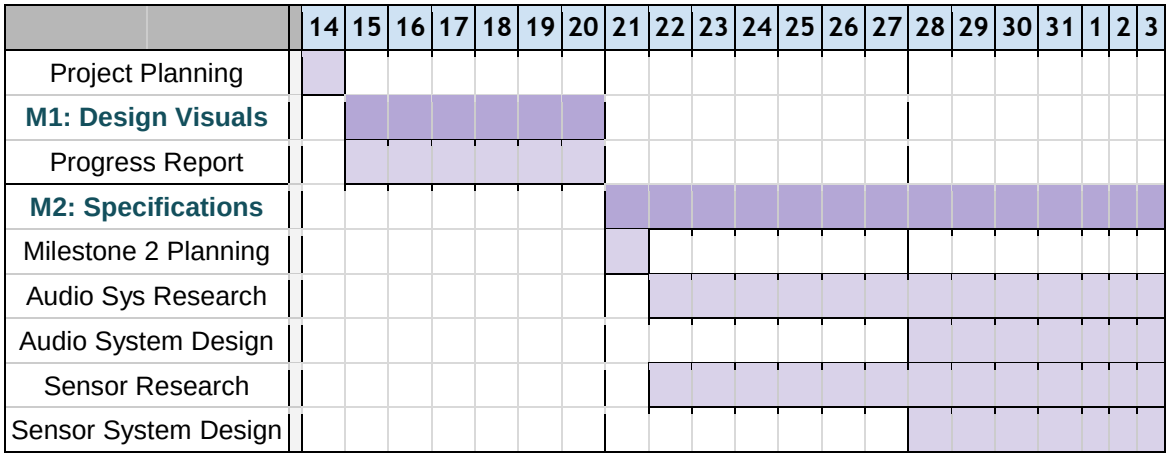
Below is the planned outline for the four milestones, due in order on October 20th, November 3rd, November 17th, and December 2nd. The intention was to finish the project and final report a week early in order to maintain a time buffer to accommodate unexpected complications. Tables 16 and 17 below outline the planned schedule in the form of a Gantt Chart.

The actual milestone completion progress follows the projected milestone tables, shown in Tables 18 and 19. It can be seen that Reading Break was not adequately anticipated and introduced a number of delays in the project.

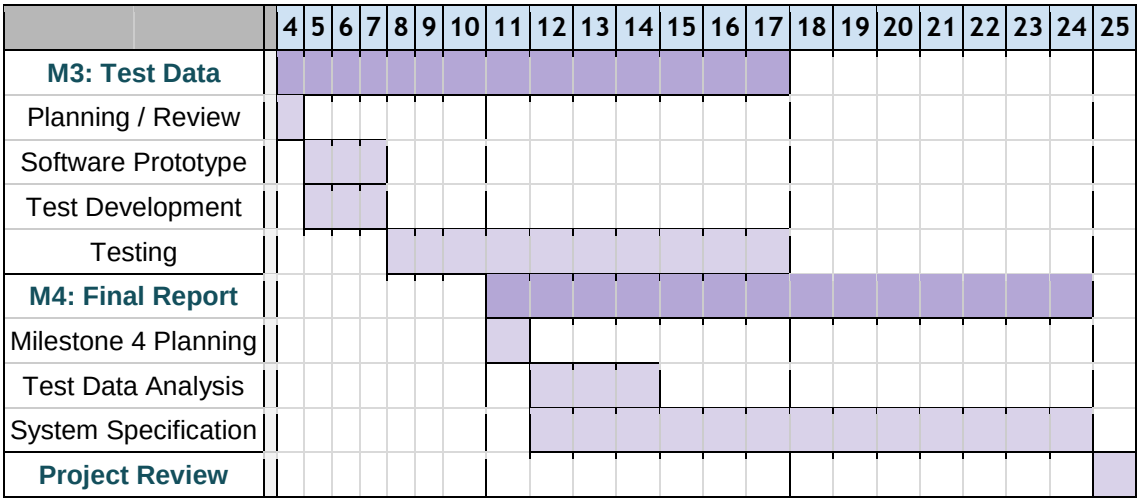
Gantt Chart Legend

Planned	Completed	Completed Late	Milestone Plan	Milestone Late

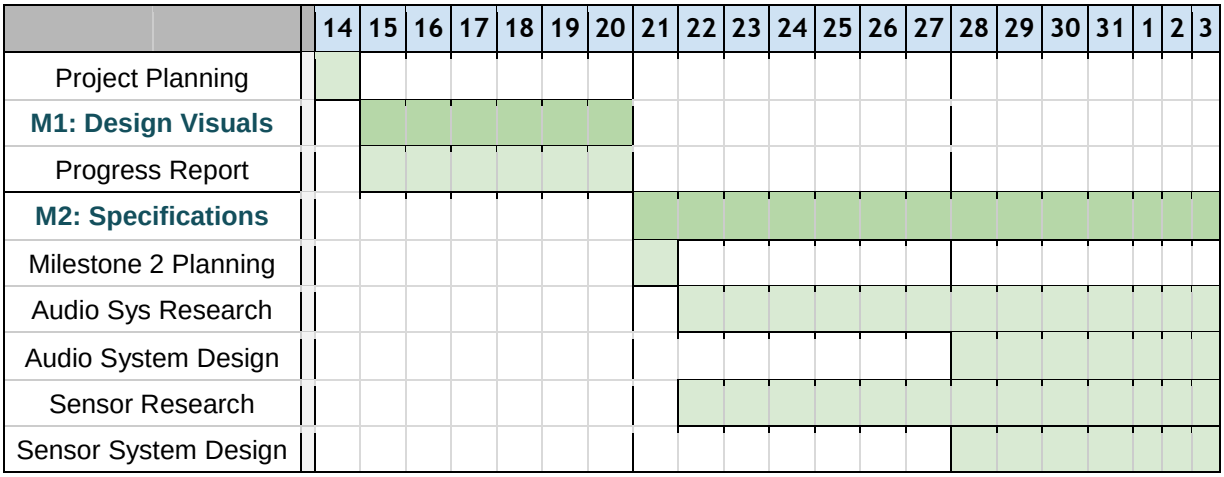
Milestone Gantt Chart Plan - Milestone 1 and 2



Milestone Gantt Chart Plan - Milestone 3 and 4



Milestone Gantt Chart Execution - Milestone 1 and 2



Milestone Gantt Chart Execution - Milestone 3 and 4