

# HEARTHERE

Infrastructure for Ubiquitous Augmented-Reality Audio

by

Spencer Russell

B.S.E., Columbia University (2008)

B.A., Oberlin College (2008)

Submitted to the Program in Media Arts and Sciences, School of Architecture and Planning, in partial fulfillment of the requirements for the degree of Master of Science in Media Arts and Sciences at the Massachusetts Institute of Technology

September 2015

© Massachusetts Institute of Technology. All rights reserved.

Author \_\_\_\_\_

Spencer Russell

MIT Media Lab

August 27, 2015

Certified by \_\_\_\_\_

Joseph A. Paradiso

Professor

Program in Media Arts and Sciences

Accepted by \_\_\_\_\_

Pattie Maes

Academic Head

Program in Media Arts and Sciences

# HEARTHERE

Infrastructure for Ubiquitous Augmented-Reality Audio

by Spencer Russell

Submitted to the Program in Media Arts and Sciences, School of Architecture and Planning on August 27, 2015, in partial fulfillment of the requirements for the degree of Master of Science in Media Arts and Sciences at the Massachusetts Institute of Technology

## *Abstract*

This thesis presents HearThere, a system to present spatial audio that preserves alignment between the virtual audio sources and the user's environment. HearThere creates an auditory augmented reality with minimal equipment required for the user. Sound designers can create large-scale experiences to sonify a city with no infrastructure required, or by installing tracking anchors can take advantage of sub-meter location information to create more refined experiences.

Audio is typically presented to listeners via speakers or headphones. Speakers make it extremely difficult to control what sound reaches each ear, which is necessary for accurately spatializing sounds. Headphones make it trivial to send separate left and right channels, but discard the relationship between the head and the rest of the world, so when the listener turns their head the whole world rotates with them.

Head-tracking headphone systems have been proposed and implemented as a best of both worlds solution, but typically only operate within a small detection area (e.g. Oculus Rift) or with coarse-grained accuracy (e.g. GPS) that makes up-close interactions impossible. HearThere is a multi-technology solution to bridge this gap and provide large-area and outdoor tracking that is precise enough to imbue nearby objects with virtual sound that maintains the spatial persistence as the user moves throughout the space. Using bone-conduction headphones that don't occlude the ears along with this head tracking will enable true auditory augmented reality, where real and virtual sounds can be seamlessly mixed.

Thesis Supervisor: Joseph A. Paradiso, Professor

# HEARTHERE

Infrastructure for Ubiquitous Augmented-Reality Audio

by Spencer Russell

*Thesis Reader*

Reader

---

Chris Schmandt  
Principal Research Scientist  
MIT Media Lab

# HEARTHERE

Infrastructure for Ubiquitous Augmented-Reality Audio

by Spencer Russell

*Thesis Reader*

Reader

---

Josh McDermott  
Assistant Professor  
MIT Department of Brain and Cognitive Sciences

## *Acknowledgments*

There have been many hands at work making this thesis possible, but in particular I'd like to thank the following people, whose contributions have been particularly invaluable.

First my advisor, Joe Paradiso, a source of great advice that I've learned to ignore at my own peril. His depth of knowledge continues to impress, and I'm grateful he believed in me enough to admit me into Responsive Environments. My readers Chris Schmandt and Josh McDermott, whose feedback and flexibility have been much appreciated. Gershon Dublon, who is so kind and generous. Without his support my user study would not have been physically or emotionally possible, and his work is a continuing inspiration. Brian Mayton is an endless supply of expertise and experience, and there's no one I'd rather have next to me in the field. Our conversations throughout this project had a tremendous impact, and his capable help in field testing was invaluable. Sang Leigh and Harshit Agrawal who were generous with their knowledge and OptiTrack system. Asaf Azaria who is always ready to help. Julian Delorme and Ken Leidel, whose dedicated work as UROPs was continually impressive. Amna, who keeps things running and is a fantastic help in navigating the Media Lab. Linda and Keira, who helped keep me on track. Jan and Bill, who kept me stocked with encouragement, corn, and bourbon, and graciously opened their home to me during my writing sojourn. My own parents Dennis and Sheryl, who have both pushed and supported me and made me who I am. Finally, my beautiful and talented wife Kate. Her support and understanding during the last few months have made this possible, and I'm looking forward to spending more time with her.

# Contents

<i>Abstract</i>	2
<i>Introduction</i>	9
<i>Related Work</i>	11
<i>Overview</i>	16
<i>Hardware</i>	18
<i>Firmware</i>	20
<i>Orientation Tracking</i>	24
<i>UWB Ranging</i>	28
<i>Bluetooth Low-Energy</i>	31
<i>iOS Application</i>	34
<i>Particle Server</i>	36
<i>GPS Integration</i>	38
<i>Calibration</i>	40
<i>Quantitative Evaluation</i>	43
<i>User Study</i>	51
<i>Conclusions and Next Steps</i>	56
<i>Appendix A: Hardware Schematic and Layout</i>	57
<i>Appendix B: Field Test Responses</i>	61
<i>References</i>	68

## *List of Figures*

1	Overview of the HearThere system components	16
2	Overview of the HearThere hardware	18
3	HearThere head tracker PCB	18
4	Firmware Architecture	20
5	IMU Axis Conventions	26
6	A Symmetric Double-Sided Two-Way Ranging exchange between the tag and an anchor	29
7	HearThere app orientation screen	34
8	HearThere app range statistics screen	34
9	HearThere app location statistics screen	35
10	HearThere app map screen	35
11	Particle server transactions	36
12	Particle system iterations	37
13	Distance calibration experimental setup	40
14	Anchor configuration for range calibration	40
15	Error data from calibration of all four anchors	41
16	Equipment for anchor calibration	41
17	Turntable configuration for calibration gyro gain	42
18	Evaluation experimental setup	43
19	Localizing A2 using optical tags	43
20	Range comparison	44
21	Overall ranging error (in cm)	44
22	Optical markers placed on the user's head for tracking	44
23	Particle filter performance	45
24	Particle filter error histogram	46
25	Orientation evaluation	47
26	Orientation evaluation detail	48

- 27 Overall error compared to Yaw/Pitch/Roll components. Note the glitches in the yaw and roll errors caused by singularities when pitch approaches  $\pm\frac{\pi}{2}$  49
- 28 Drift evaluation 50
- 29 Field test area 51
- 30 Field test localization error 54
- 31 Localization error grouped by scene 54
- 32 Localization error grouped by headphone type 54
- 33 Localization error grouped by whether the source was covered by UWB or not 54
- 34 Field test tracking data 55



# Introduction

*Alice is driving, wearing her auditory display headset, a lightweight device that goes around the back of her neck with small transducers on the bones just in front of her ears. Based on her GPS location and head orientation, the system synthesizes a gentle tone that moves from her current position along the road to her next turn, where it emits a soft ping and after turning right fades out, then starts again from her current position. As she approaches the turn the navigation sound gradually becomes slightly louder, as well as more frequent (because it takes less time to get from her current position to the turn). While driving Alice is peripherally aware of the cars around her because she can hear them, though with her windows up and music on she can't hear them directly. She is hearing the cars through her auditory display, spatialized to their actual positions which are picked up by her car's sensors. She briefly wonders why her navigation system is taking a different route than normal, but on closer listening recognizes the sound of construction ahead in the distance, far beyond her real-world perceptual abilities.*

*Alice arrives at TidMarsh, a wetland restoration site that is blanketed with environmental sensors. Ultra-wideband base stations localize her headset with much higher precision than GPS. Alice is now walking through a responsive musical composition that responds to the sensor data in real-time, surrounding her with a shifting soundscape. A microphone in a near-by tree catches the sound of a family of birds, which Alice hears as if through an audio telescope. As she walks past the microphones and sensors she hears their sound moving around her realistically, seamlessly blending to wind and birdsong she hears through her unobstructed ears. Hearing a melody she particularly enjoys coming from a nearby sensor she gets close to it to hear more clearly.*

While advances in computational power have enabled the use of real-time head-related transfer functions (HRTFs) and room modeling to create more realistic sonic environments, the illusion of presence is immediately shattered when the user moves their head and hears the whole sonic world move with them.

We believe that spatial audio that maintains registration with the real world creates a less intrusive and more compelling auditory display. Further, we consider head-tracking to be an integral component in any spatialized audio system that attempts to fuse virtual sounds with the user's natural auditory environment for

two related reasons. Firstly, researchers have established that self-motion is important for resolving front-back confusion errors<sup>1</sup> that are otherwise difficult to overcome. Additionally head tracking is important simply because a user's head is often not aligned with their body, so using body orientation (e.g. from the mobile orientation in a shirt pocket) as in Blum et al.<sup>2</sup> causes large perceptual errors whenever the user moves their head independently from their torso. In addition to the well-established localization benefits, we believe that head tracking has an important impact on perceived realism, though that area is under-explored in the literature as most research only measures the subjects' accuracy in localization tasks, not the accuracy of the illusion.

Advances in MEMS sensing now allow a full 9DOF IMU (3-axis each of accelerometer, gyroscope, and magnetometer) in a low-power IC package, and there are many sensor fusion algorithm implementations available.<sup>3</sup> The wider adoption of Ultra WideBand (UWB) RealTime Localization Systems (RTLS) has also driven costs down and availability up of precise localization with chips available at low cost in small quantities. To take advantage of these advancements we have created HearThere, a multi-scale head-tracking audio system that uses UWB localization anchors when available, but gracefully falls back to GPS when outside UWB range.

This work is distinct from the work described in the *Related Work* chapter in that it can scale to many users over a large geographic area. Within the UWB zones the system can represent audio sources around the user with sub-meter accuracy, and the designed experience can span many of these zones. This work is also unusual in that we are committed to preserving the user's experience of their natural surroundings, and show that bone conduction is a viable technology to present spatial audio to users without occluding their ears.

<sup>1</sup> Durand R. Begault, Elizabeth M. Wenzel, and Mark R. Anderson. "Direct comparison of the impact of head tracking, reverberation, and individualized head-related transfer functions on the spatial perception of a virtual speech source". 10 (2001).

<sup>2</sup> Jeffrey R. Blum, Mathieu Bouchard, and Jeremy R. Cooperstock. "What's around me? Spatialized audio augmented reality for blind users with a smartphone". 2012.

<sup>3</sup> OlliW. *IMU Data Fusing: Complementary, Kalman, and Mahony Filter*. [Online; accessed 5-November-2014]. Sept. 2013.

# Related Work

## Indoor Localization

Because of the abundance of applications for precise indoor localization, it is a very active research area with many possible approaches, and there are a variety of commercial products available. **Mautz**<sup>4</sup> provides one of the most recent surveys of available techniques. The survey categorizes and compares the technologies as well as evaluating their fitness for a variety of use-cases. **Hightower and Borriello**<sup>5</sup> also describe many of the foundational works in the field and include a well-developed taxonomy. At a high level most of these technologies can be categorized along two axes: the signal used (optical, RF, acoustic, etc.) and the properties of the signal used for localization (time-of-arrival, angle-of-arrival, signal strength, etc.).

Optical tracking systems are currently popular, particularly commercial systems such as **OptiTrack**<sup>6</sup> from NaturalPoint and the **Vicon**<sup>7</sup> system. These camera-based optical systems operate by transmitting infrared (IR) light that bounces off of retro-reflective markers on the objects to be tracked. While these systems support precision on the order of millimeters, one main downside is that they have no way to distinguish individual markers. Groups of markers must be registered with the system prior to tracking, and the system can easily get confused if the marker groups have symmetries that prevent it from uniquely determining an orientation.

Other optical systems such as the **Prakash**<sup>8</sup> system and **ShadowTrack**,<sup>9,10</sup> replace the cameras with infrared projectors and use simple photo receptors as tags. **Prakash** tags the capture volume with grey-coded structured light from multiple simple projectors, while **ShadowTrack** has a rotating cylindrical film around the light source with spread-spectrum inspired coding, and the tags can determine their angle by cross-correlating the transmission signal with one from a fixed receptor.

Recently Valve Corporation has introduced their **Lighthouse**

<sup>4</sup> Rainer Mautz. "Indoor positioning technologies". Habilitation Thesis. 2012.

<sup>5</sup> Jeffrey Hightower and Gaetano Borriello. "Location systems for ubiquitous computing". 8 (2001).

<sup>6</sup> <http://www.optitrack.com/>

<sup>7</sup> <http://www.vicon.com/>

<sup>8</sup> Ramesh Raskar et al. "Prakash: lighting aware motion capture using photosensing markers and multiplexed illuminators". 2007.

<sup>9</sup> Karri T. Palovuori, Jukka J. Vanhala, and Markku A. Kivikoski. "Shadow-track: A Novel Tracking System Based on Spread-Spectrum Spatio-Temporal Illumination". 6 (Dec. 2000).

<sup>10</sup> I. Mika et al. "Optical positioning and tracking system for a head mounted display based on spread spectrum technology". 1998.

system which scans a laser line through the tracked space, similar to the iGPS system from Nikon Metrology, described by Schmitt et al.<sup>11</sup> Any system based on lasers or projection and intended for user interaction faces a trade-off between performance in bright ambient light (such as sunlight) and eye-safety issues, though the signal-to-noise ratio can be improved significantly by modulating the signal. There is little public documentation or rigorous evaluation of the Lighthouse system, so we look forward to learning more about it.

**SLAM** (Simultaneous Location and Mapping) is a camera-based optical approach that places the camera on the object to be tracked. This approach is attractive because it does not require any infrastructure to be installed, but it does require heavy processing in the tag.

Geometric approaches based on audible or ultrasonic sound waves have much less strict timing requirements when compared to RF approaches because sound moves about six orders of magnitude slower. Unfortunately the speed of sound varies substantially with temperature and is affected by wind, as well as subject to dispersive effects of the air. Temperature variations in outdoor spaces are often too large to be compensated for with measurements at the endpoints.

Several radio-frequency (RF) (including UWB) and electromagnetic approaches are also common in this design space. Systems with transmitters often use a geometric approach including triangulation (angle-based) or trilateration (distance-based). Other systems attempt to make use of signals already in the air.

**Chung et al.**'s geomagnetic tracking system<sup>12</sup> builds a database of magnetic field distortions and then at runtime attempts to locate the tag by finding the most similar database entry. This is a general approach known as fingerprinting which has also been widely explored to use ambient WiFi signals, particularly because geometric approaches using ambient signals have proven difficult due to field distortions and multi-path effects. Fingerprinting requires an often-exhaustive measurement process of the space to be tracked, and reducing or eliminating this step is an active research area. Accuracy of these methods tends to be on the order of 1 or more meters.

Ultra-WideBand (UWB) is a popular approach for geometric tracking because it enables much more precise time-of-flight (ToF) measurements due to the short pulse length and sharp transitions (see the *UWB Ranging* chapter for more details). Previous work<sup>13,14</sup> has investigated combining GPS and UWB to cover both outdoor and indoor localization with promising results.

<sup>11</sup> R. Schmitt et al. "Performance evaluation of iGPS for industrial applications". Sept. 2010.

<sup>12</sup> Jaewoo Chung et al. "Indoor location sensing using geo-magnetism". 2011.

<sup>13</sup> Jose Gonzalez et al. "Combination of UWB and GPS for indoor-outdoor vehicle localization". 2007.

<sup>14</sup> David S. Chiu and Kyle P. O'Keefe. "Seamless outdoor-to-indoor pedestrian navigation using GPS and UWB". 2008.

In particular Chiu et al. show UWB performs favorably even to high-precision commercial Code DGPS outdoors. Their focus was more on validating the basic technology in controlled laboratory environments using off-the-shelf UWB and GPS equipment and did not attempt to build an integrated system that could be used by the general public. They also do not evaluate highly-dynamic motion, instead stopping periodically at waypoints with known locations.

### *Location-Based Sound and Augmented Reality Audio*

**Azuma**<sup>15</sup> Provides a simple and useful definition of Augmented Reality, which is that it

- Combines real and virtual
- Is interactive in real time
- Is registered in 3-D

The third criteria is useful for separating *Augmented Reality Audio* (ARA) from *Location-Based Sound* (LBS). The key difference is that in LBS the sound cannot be said to be registered to a particular location in 3D space. For example, **Audio Aura**<sup>16</sup> is a location-based sound system in an office environment, but not augmented reality audio because the sounds are simply triggered by the user's location and played through headphones. **ISAS**<sup>17</sup> presents spatialized audio content with a defined location in 3D space, but uses the user's mobile to determine orientation. While registration is attempted, it is relatively coarse. They provide some helpful insights into sound design in an assistive context, and demonstrate a simple spatializer that models ITD, ILD, and applies a lowpass filter to simulate ear occlusion effects for sources behind the user. Their application was able to spatialize up to four simultaneous sources.

Wu-Hsi Li's **Loco-Radio**<sup>18</sup> uses a mobile phone mounted to the user's head for orientation tracking, and provides insight into allowing the user to zoom in and out of the scene, changing the radius of their perception. Using the location tracker from Chung et al. they had a location precision of about 1 m, updated at 4 Hz.

**LISTEN**<sup>19</sup> is an ARA system including an authoring system. The project focuses on context-awareness and providing content based on individualized profiles and inferences based on the user's behavior, such as how they move through the space and where they direct their gaze.

At SIGGRAPH 2000, AuSIM Inc. presented **InTheMix**,<sup>20</sup> which

<sup>15</sup> Ronald T. Azuma et al. "A survey of augmented reality". 4 (1997).

<sup>16</sup> Elizabeth D. Mynatt et al. "Designing audio aura". 1998.

<sup>17</sup> Blum, Bouchard, and Cooperstock, "What's around me?"

<sup>18</sup> Wu-Hsi Li. "Loco-Radio: designing high-density augmented reality audio browsers". PhD thesis. 2013.

<sup>19</sup> Andreas Zimmermann, Andreas Lorenz, and S. Birlinghoven. "LISTEN: Contextualized presentation for audio-augmented environments". 2003.

<sup>20</sup> W. L. Chapin. "InTheMix". 2000.

presented responsive musical content. The music was spatialized using HRTFs as well as room modeling, and their system integrated with a number of commercial tracking systems to track the user's head. The experience was limited to a 4 m radius circle, and the user was tethered for audio and tracking purposes.

In the assistive space, **SWAN**<sup>21</sup> is a backpack-sized audio-only navigation and wayfinding system that uses bone-conduction headphones. It uses commercial GPS receivers for location and either a digital compass or an off-the-shelf 9-axis IMU, updating at 30 Hz. Blind users have apparently been successful navigating with the system, but they do not give any hard metrics that would be useful for comparison. They also do not specifically address issues particular to spatial audio delivered over bone conduction.

### *Spatial Audio Delivery and Perception*

ARA systems often use standard in-ear or over-ear headphones, which interferes with the user's perception of the world around them. **Härmä et al.** present a system<sup>22</sup> that includes what they refer to as *hear-through* headphones integrate binaural microphone capsules into a pair of in-ear headphones. They evaluate their work with laboratory listening tests. This work is one of a few to investigate the extent to which users can distinguish between real and virtual sounds, and in some cases their subjects have a difficult time distinguishing. The "real" sounds are in this case mediated through the microphone/headphone device though, so it is impossible to distinguish whether the confusion is due to the quality of the virtual sound spatialization or degraded spatialization of external sounds.

It has been shown that head motion plays an important role in our ability to localize sound,<sup>23</sup> particularly in reducing front/back confusion errors. Though some results<sup>24</sup> have found less compelling evidence, and no correlation with externalization. Brimijoin and Akeroyd modernized Wallach's approach<sup>25</sup> and showed that as the test signal bandwidth goes from 500 Hz to 8 kHz, spectral cues become as important head movement cues (in situations where they are contradictory). For our purposes it's important to keep in mind that even experiments that don't allow head movement assume that the head orientation is *known*. Without knowing the orientation the system is simply guessing. Because of this head tracking is a requirement.

For navigation tasks spatialized audio has shown to create lower cognitive load than spoken directions.<sup>26</sup>

There have also been several commercial products that have

<sup>21</sup> Jeff Wilson et al. "Swan: System for wearable audio navigation". 2007.

<sup>22</sup> Aki Härmä et al. "Augmented reality audio for mobile and wearable appliances". 6 (2004).

<sup>23</sup> Hans Wallach. "The role of head movements and vestibular and visual cues in sound localization." 4 (1940); Willard R. Thurlow and Philip S. Runge. "Effect of induced head movements on localization of direction of sounds". 2 (1967); Pauli Minnaar et al. "The importance of head movements for binaural room synthesis" (2001).

<sup>24</sup> Begault, Wenzel, and Anderson, "Direct comparison of the impact of head tracking, reverberation, and individualized head-related transfer functions on the spatial perception of a virtual speech source".

<sup>25</sup> W Owen Brimijoin and Michael A Akeroyd. "The role of head movements and signal spectrum in an auditory front/back illusion". 3 (2012).

<sup>26</sup> Roberta L. Klatzky et al. "Cognitive load of navigating without vision when guided by virtual sound versus spatial language." 4 (2006).

added head tracking to headphones for virtual surround sound:

- DSPeaker HeadSPeaker
- Smyth Research Realiser A8
- Beyerdynamic DT 880 HT
- Sony VPT

Latency has an effect on our ability to localize.<sup>27</sup> Azimuth error was shown to be significantly greater at 96ms latency than 29ms, and latency had a greater effect than update rate or HRTF measurement resolution. This provides some guidelines and target values to shoot for.

Bone conduction headphones have limited bandwidth, which reduces the audio quality for full-spectrum sources like music. It also presents challenges when presenting users with spectral location cues such as HRTFs. Several studies have tried to measure these issues, though none have been particularly conclusive. MacDonald et al.<sup>28</sup> found that localization performance using the bone conduction headphones was almost identical to a pair of over-ear headphones. The sources were filtered to fit within the bandwidth of the BC headphones, and their measurements were very coarse-grained (only in 45° increments) though, so it only proves suitability for very basic localization. As part of the SWAN project, Walker et al. evaluated navigation performance when following spatialized audio beacons using bone conduction headphones.<sup>29</sup> While performance was somewhat degraded from previous work with normal headphones, the study at least confirms that unmodified HRTFs presented through bone conduction headphones can perform basic spatialization.

<sup>27</sup>J. Sandvad. "Dynamic Aspects of Auditory Virtual Environments". May 1996.

<sup>28</sup>Justin A. MacDonald, Paula P. Henry, and Tomasz R. Letowski. "Spatial audio through a bone conduction interface: Audición espacial a través de una interfase de conducción ósea". 10 (Jan. 2006).

<sup>29</sup>Bruce N. Walker and Jeffrey Lindsay. "Navigation performance in a virtual environment with bonephones" (2005).

# Overview

The HearThere system is comprised of four main components as diagrammed in Figure 1.

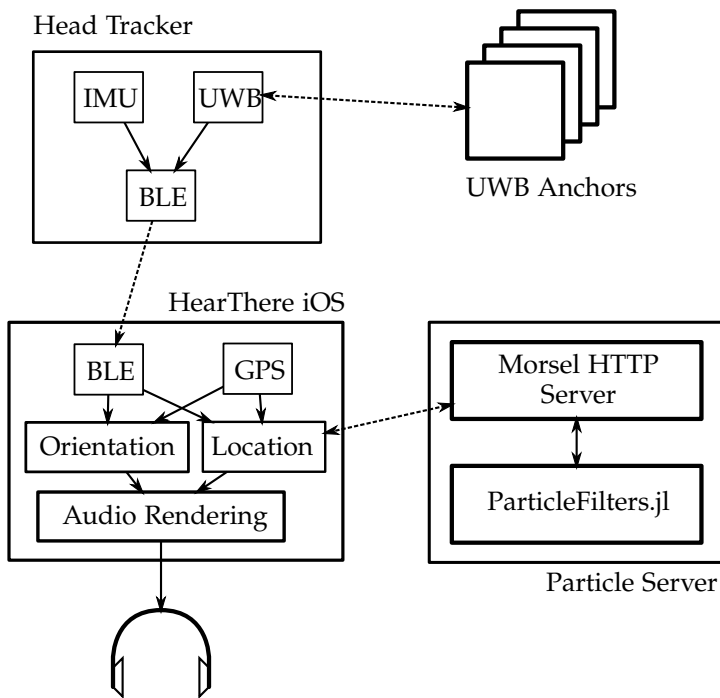


Figure 1: Overview of the HearThere system components

The head tracker hardware is worn on the user's head and communicates with a mobile phone or laptop over Bluetooth Low-Energy (BLE). It is responsible for maintaining an orientation estimate using its internal inertial measurement unit (IMU) as well as performing 2-way ranging exchanges with each of the four Ultra-WideBand anchors, which are EVB1000 evaluation boards from DecaWave.

The HearThere iOS application is built in the Unity3D game



engine and maintains a 3D audio scene that is played to the user through a set of commercial bone-conduction headphones. Audio output uses the usual headphone output, so switching playback equipment is easy. As the user moves around in their environment, the application uses GPS as well as the location and orientation data from the head tracker hardware and moves a virtual listener in the game environment to match the user's movements, creating a virtual audio overlay. The application can also send the data over the network using the Open Sound Control Protocol (OSC) which is useful for data capture and logging or for integrating with other real-time systems.

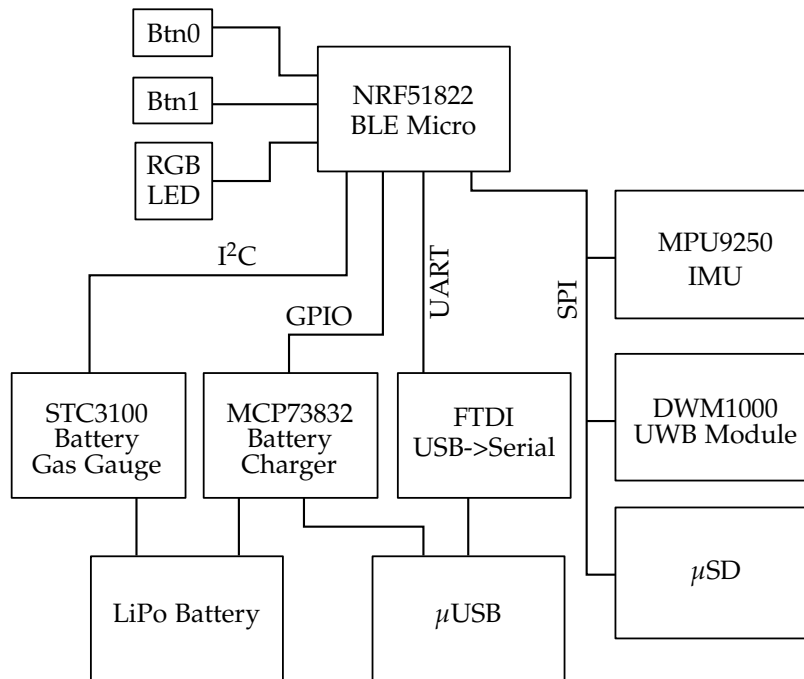
When the user is within range of the UWB anchors, the software sends the ranges measured by the head tracker hardware to the Particle Server in the payload of an HTTP request. The Particle Server is built in the Julia<sup>30</sup> programming language, and includes an HTTP front-end server using the Morsel HTTP library<sup>31</sup> and a custom particle filter implementation. The Particle Server sends a location estimate to the application in the HTTP response, and also includes the standard deviation vector of the particle mass as an indication of confidence.

<sup>30</sup> Jeff Bezanson et al. "Julia: A Fresh Approach to Numerical Computing" (Nov. 2014).

<sup>31</sup> <https://github.com/juliaweb/morsel.jl>

# Hardware

The HearThere HeadTracking hardware (pictured in Figure 3) is designed in a development board form-factor and optimized for development and testing. The schematic and layout are included in Appendix A.



The main microcontroller is the nRF51822 from Nordic Semiconductor, which also handles communication with the host over Bluetooth Low-Energy. It communicates with the InvenSense MPU-9250 IMU over the SPI bus, as well as the DecaWave DWM1000 Ultra-WideBand (UWB) module. It includes a reset button, two general-purpose buttons, and an RGB LED for user feedback. The

Figure 2: Overview of the HearThere hardware

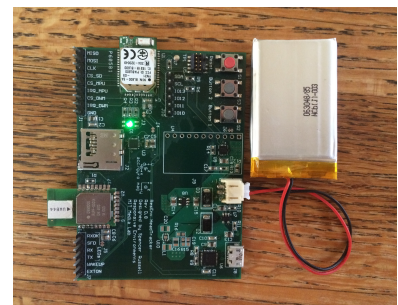


Figure 3: HearThere head tracker development edition. This version is optimized for ease of development rather than board size, and future versions will be considerably smaller.

user can switch between indoor (no magnetometer) and outdoor (with magnetometer) modes by holding down Button 1 during boot, and the device indicates its mode by flashing red or green for indoor and outdoor mode, respectively. To enable data logging applications at higher resolution than can fit in the BLE bandwidth, the board has an on-board SD card slot for local data capture.

The board is powered by a LiPo battery, which can be charged via the micro-USB port. There is a battery monitoring chip that measures the battery voltage and also integrates current in and out to estimate charge. This data can be reported to the nRF51822 over I<sup>2</sup>C. The SPI and I<sup>2</sup>C communication busses are broken out to 0.100" headers, as well as all select and IRQ lines. This makes debugging with a logic analyzer much more convenient. Communication to the SD card and battery monitor chip is not yet implemented in the firmware.

# Firmware

## Architecture

The firmware for the HearThere head tracker is written in C and runs on the nRF51822 chip from Nordic Semiconductor, which is built around an ARM Cortex-M0. The code is broken into modules, each of which consists of a header (.h) file and a source (.c) file. Figure 4 depicts the major modules, and omits some less important utility modules for clarity.

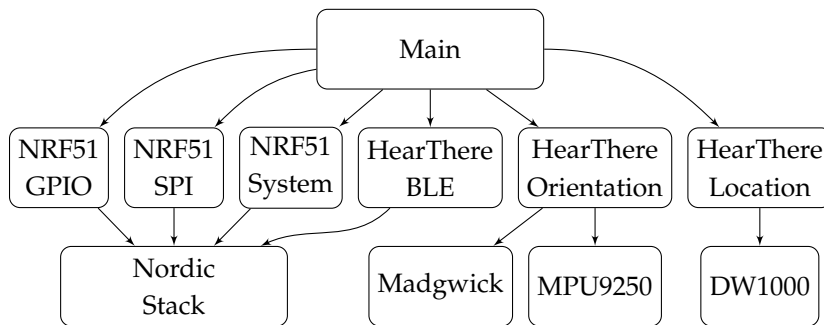


Figure 4: Firmware Architecture

The main responsibilities of each module are:

**Main** initializes the hardware and the top-level modules, and supplies each module with a reference to any hardware peripherals it needs.

**NRF51System** handles System-wide functionality such as disabling interrupts and handling assertion failures.

**NRF51GPIO** initializes and manages GPIO pins.

**NRF51SPI** initializes and manages the SPI peripherals, and provides an interface for reading and writing from them.

**HearThereOrientation** reads sensor data from the IMU and performs the sensor fusion. It takes a reference to a SPI device that it uses to communicate with the IMU, and calls a callback when new orientation data is available.

**HearThereLocation** reads range data from the anchors. It also takes a reference to a SPI device that is used to communicate with the DW1000 chip from DecaWave. It calls a callback when new range data is available.

**HearThereBLE** uses the Nordic BLE API to manage the BLE connection to the host. It provides an API for sending sensor data and calls a callback on connection status changes.

**MPU9250** initializes and reads from the MPU9250 9-axis IMU from InventSense.

**Madgwick** runs the sensor fusion state machine, based on an algorithm and code from Sebastian Madgwick.<sup>32</sup>

**DW1000** wraps the provided DecaWave library and provides an API for UWB ranging.

<sup>32</sup> Sebastian OH Madgwick. “An efficient orientation filter for inertial and inertial/magnetic sensor arrays” (2010).

### *Peripheral Drivers*

You can see in Figure 4, only the Main module depends on the platform-specific hardware drivers such as NRF51SPI and NRF51GPIO. Main is responsible for initializing this hardware, and it passes pointers to the driver’s description structs into any modules that need access. For instance, the HearThereOrientation module needs to use the SPI driver to talk to the IMU.

Rather than including the NRF51SPI header, the HearThereOrientation module includes a generic header SPI.h, and pointers are cast to a generic SPI struct type. This makes the code much more portable and decoupled from the hardware implementation, as we can re-use these modules on a completely different platform simply by implementing a new set of drivers that satisfy the same interface.

### *Interfaces to Vendor Code*

Along with hardware, vendors often provide source or binary software that can be included in a project to speed up development. In HearThere, we use DecaWave’s low-level API code that provides functions corresponding to the various SPI commands that the DW1000 chip supports. Fortunately they provide an easy way to incorporate their code into your project by supplying functions with predefined names to access the SPI peripheral. The DecaWaveShim module provides these functions wrapped around our SPI driver.

Our peripheral drivers rely on the Nordic stack for the NRF51822 chip to provide access to hardware and the HearThereBLE module

uses their BLE API directly, as well as some useful stack features for scheduling. For the most part we have successfully isolated this platform-specific code from the main functional modules, but in a few cases (e.g. endianness conversion macros) some platform-specific code has crept in.

## *Scheduling*

HearThere uses a simple cooperative task scheduling design, in which each module has a tick function that is called from the main loop. Each module is responsible for maintaining their own state machine and in general the modules avoid busy-waiting so that other tasks can run.

Minimizing latency was a driving design factor, and one of the tightest latency deadlines came from managing the UWB ranging process (see the *UWB Ranging* chapter for more details), where we need to deterministically handle the incoming UWB message and finish responding within 2 ms. The Nordic BLE stack interrupts the application for 1 ms for each connection (in our case approximately every 20 ms), and the process of responding takes about 600  $\mu$ s, so we can tolerate a maximum handling latency of 400  $\mu$ s. Unfortunately running one iteration of the Madgwick sensor fusion algorithm (see page 24) takes 2.8 ms (the nRF51822 does not have a floating-point unit). One option would be to handle the UWB communication in an interrupt context, but this is complicated by the need to use the SPI peripheral to communicate with the DW1000 chip, which could corrupt other in-progress communications. We elected to re-write the Madgwick code into a state-machine so that the computations could be done in smaller pieces of less than 400  $\mu$ s each.

After measuring the various tasks running on the system we estimated that without partitioning the fusion calculations we would need to slow down our ranging rate to under 3 Hz to make our deadlines. With partitioning we estimated we could run at 16.7 Hz, and in practice we were able to get 15 Hz. All tests were run while reading from the IMU and updating the sensor fusion algorithm at 200 Hz, and sending updated orientation over BLE at approximately 35 Hz to 40 Hz. In later experiments the Anchor range update rate was reduced to 7 Hz to 10 Hz to ensure more reliable operation due to more timing headroom.

## *SoftDevice*

It's common in embedded RF development to link the application code against a binary library provided by the vendor that provides low-level RF functionality, commonly referred to as the vendor's *stack*. This can lock the developer into using the same compiler as the vendor, as the developer needs to link her application code against the stack. The Nordic architecture however, builds the stack code as what they call a *SoftDevice*. Rather than linking the stack and application during the build, the SoftDevice is flashed to a separate region of the chip's flash memory. All calls from the application to the SoftDevice's API are mediated through the Supervisor Call (SVC) instruction that's part of the instruction set used by the ARM Cortex-M0.<sup>33</sup>

Typically this feature is used for user-space code to trigger operating-system code, and takes an 8-bit value which is passed to a software interrupt that is handled by the OS. This decoupling of application and stack development is particularly useful in an RF development context, as we can use any compiler and toolchain that supports the ARM. In practice there is some compiler-specific code in the Nordic-provided header files, but fortunately they support several common compilers, including GCC, which we used for our development.

<sup>33</sup> Joseph Yiu. *The Definitive Guide to the ARM Cortex-M0*. Apr. 2011.

## Orientation Tracking

The HearThere head tracker relies on a MEMS inertial measurement unit(IMU) chip from InvenSense called the MPU-9250. It provides a 3-axis gyroscope (measures angular velocity), 3-axis accelerometer (measures a combination of gravity and translational acceleration), and 3-axis magnetometer (measures the local magnetic field vector).

When the device is motionless (or moving at constant velocity) the accelerometer measurement should be entirely due to gravity, which gives a clue as to the orientation (tells us which way is up), but leaves ambiguity because of possible rotation about that vector. The magnetic field vector however, is linearly independent and so the two combined should give a unique orientation solution.

Under acceleration however, we can't be sure which part of the measured acceleration is due to gravity. This is where the gyroscope comes in, because if we have a previous known orientation we can integrate our rotational velocity to get a new orientation relative to the old one. So if we know our initial orientation (for instance by reading the accelerometer and magnetometer on bootup), in principle we can rely on the gyroscope to update our orientation.

In practice this method is hindered by gyroscope noise, which after integration becomes a random walk that causes our orientation estimate to gradually drift. The search for methods for correcting this drift by combining the available sensor data (sensor fusion) has been an active research area dating at least to the inertial guidance system development of the mid-twentieth century,<sup>34</sup> and common approaches include complementary filters, extended Kalman filters, and unscented Kalman filters. In general the approach is to integrate the gyroscope to capture short-term variations, while taking into account the magnetometer and accelerometer data over longer time periods to compensate for the gradual drift. For indoor environments, even the magnetometer is often not reliable due to sources of electro-magnetic interference as well as magnetic field distortion caused by nearby metal. In

<sup>34</sup> Donald A. MacKenzie. *Inventing accuracy: A historical sociology of nuclear missile guidance*. 1993.



these cases, it is often best to simply disable the magnetometer, though this results in drift around the gravity vector. In the future we would like to explore alternative approaches to this drift compensation, including comparing the integrated accelerometer data with position data from the localization system. This should help enable drift-free operation even when without the benefit of the magnetometer.

### *Sensor Fusion*

HearThere uses the Madgwick algorithm<sup>35</sup> based on prior success<sup>36</sup> and the availability of efficient C-language source code that could be run on our microcontroller.

The output of the Madgwick algorithm provides a quaternion relating an estimated global coordinate frame to the coordinate frame of the IMU. What we are actually interested in though, is the quaternion relating the actual earth frame to the user's head. To make this transformation we introduce the four relevant coordinate frames: Earth, Estimated Earth, Sensor, and Head, and the quaternions that relate each frame to the next.  ${}_{\text{Est}}^{\text{Earth}}q$  relates the actual earth frame to where the Madgwick algorithm thinks it is. This transformation represents an error that could be due to a lack of magnetic reference (as when we are operating in indoor mode), or error due to magnetic declination.  ${}_{\text{Sensor}}^{\text{Est}}q$  is the quaternion reported by the Madgwick algorithm.  ${}_{\text{Head}}^{\text{Sensor}}q$  relates the sensor to the user's head, based on how it is mounted or worn. Here we use the notation  ${}_{\text{B}}^{\text{A}}q = (q_0 \ q_1 \ q_2 \ q_3) = q_0 + q_1\mathbf{i} + q_2\mathbf{j} + q_3\mathbf{k}$  to be the quaternion going from frame  $A$  to  $B$ . The  $\mathbf{i}$ ,  $\mathbf{j}$ , and  $\mathbf{k}$  terms correspond to the  $x$ ,  $y$ , and  $z$  axes, respectively. See Kuipers<sup>37</sup> for a more complete treatment of quaternions.

Because we want the overall transformation from Earth to Head, we set

$${}_{\text{Head}}^{\text{Earth}}q = {}_{\text{Est}}^{\text{Earth}}q \times {}_{\text{Sensor}}^{\text{Est}}q \times {}_{\text{Head}}^{\text{Sensor}}q \quad (1)$$

Our iOS application has a *ReZero* button that the user presses while looking directly north with a level head to determine  ${}_{\text{Est}}^{\text{Earth}}q$ . This operation typically happens at the beginning of an interaction, but can be repeated if the orientation estimate drifts. The position of looking due north aligns the users head coordinate frame with Earth's, so

<sup>35</sup> Sebastian OH Madgwick, Andrew JL Harrison, and Ravi Vaidyanathan. "Estimation of IMU and MARG orientation using a gradient descent algorithm". 2011.

<sup>36</sup> Brian Mayton. "WristQue: A Personal Sensor Wristband for Smart Infrastructure and Control". MA thesis. 2012.

<sup>37</sup> Jack B. Kuipers. *Quaternions and rotation sequences*. 1999.

$$\begin{matrix} \text{Earth} \\ \text{Head} \end{matrix} q = (1 \ 0 \ 0 \ 0) \quad (2)$$

$$\begin{matrix} \text{Earth} \\ \text{Est} \end{matrix} q \times \begin{matrix} \text{Est} \\ \text{Sensor} \end{matrix} q \times \begin{matrix} \text{Sensor} \\ \text{Head} \end{matrix} q = (1 \ 0 \ 0 \ 0) \quad (3)$$

$$\begin{matrix} \text{Est} \\ \text{Sensor} \end{matrix} q \times \begin{matrix} \text{Sensor} \\ \text{Head} \end{matrix} q = \begin{matrix} \text{Earth} \\ \text{Est} \end{matrix} q^* \quad (4)$$

$$\left( \begin{matrix} \text{Est} \\ \text{Sensor} \end{matrix} q \times \begin{matrix} \text{Sensor} \\ \text{Head} \end{matrix} q \right)^* = \begin{matrix} \text{Earth} \\ \text{Est} \end{matrix} q \quad (5)$$

Where  $\times$  is quaternion multiplication and  $q^*$  indicates the conjugate of  $q$ , or  $(q_0 \ -q_1 \ -q_2 \ -q_3)$ . Also note that because the set of unit quaternions double-covers the set of 3D rotations,  $(q_0 \ q_1 \ q_2 \ q_3)$  represents the same rotation as  $(-q_0 \ -q_1 \ -q_2 \ -q_3)$ .

We assume that  $\begin{matrix} \text{Sensor} \\ \text{Head} \end{matrix} q$  is known because we know how the sensor is mounted, so the above expression allows us to calculate  $\begin{matrix} \text{Earth} \\ \text{Est} \end{matrix} q$ , which we can plug into eq. 1 every sensor frame to get the overall head orientation relative to the Earth frame. This happens in the iOS application.

### Axes and Handedness

When working in 3D systems it is common to deal with data, algorithms, or documentation that assumes different axis conventions than your system. An axis convention can be described by the physical interpretation of each of the three axes ( $x$ ,  $y$ , and  $z$ ), as well as the direction of positive rotation (right-handed or left-handed). For instance, in the accelerometer of the MPU-9250 (pictured in Figure 5) the  $x$  axis points to the right, the  $y$  axis points up, the  $z$  axis points out the front of the PCB. This coordinate system can be said to be *right-handed* because if you place the thumb, and index fingers of your right hand along the  $x$  and  $y$  axes respectively, your middle finger will point along the  $z$  axis. Notice this also works for any rotation of the axes, i.e.  $yzx$  and  $zxy$ .

The gyroscope follows a slightly different right-hand rule, where if you place your thumb along the axis of rotation, your fingers curl in the direction of positive rotation. Notice in Figure 5 that the magnetometer uses a left-handed coordinate system, requiring a change of axis in the firmware. Though there are two handedness rules (one for rotation and one for axis relations), they are related in that the handedness of the axes should always match the handedness of rotation, in order for cross-products and quaternion math to work as expected.

The published implementation of the Madgwick algorithm uses a North-East-Down or NED axis convention, so  $x$  points North,  $y$  points East, and  $z$  points Down. This is a right-handed coordinate

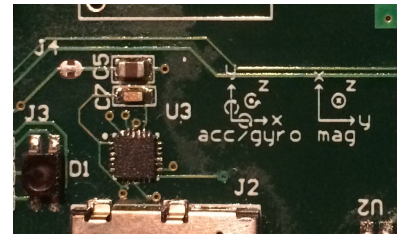


Figure 5: Portion of the HearThere head tracker PCB showing the axis conventions used by the MPU-9250 IMU (U3)

system, so we chose to standardize the IMU axes in a right-handed system as well by swapping the  $x$  and  $y$  axes of the magnetometer. The Unity3D game engine that we used to implement our iOS application uses a left-handed coordinate system where  $x$  points East,  $y$  points Up, and  $z$  points North.

	$x$	$y$	$z$	Hand
Madgwick	North	East	Down	Right
Unity	East	Up	North	Left

Table 1: Coordinate system summary

So to take a vector  $v_m = [x_m, y_m, z_m]$  in the Madgwick space and convert to Unity space gives  $v_u = [y_m, -z_m, x_m]$  and a quaternion  $q_m = (q_{m0} \ q_{m1} \ q_{m2} \ q_{m3})$  in Madgwick space becomes  $q_u = (q_{m0} \ -q_{m2} \ q_{m3} \ -q_{m1})$  in Unity space. Our firmware transmits the quaternion in Madgwick axes and it is converted to Unity axes in the iOS application.

# UWB Ranging

Ultra-WideBand or UWB, describes the use of RF signals with an absolute bandwidth of greater than 500 MHz or relative bandwidth greater than 20%.<sup>38</sup> After a period of initial development in the 1960s and 1970s<sup>39</sup> there was a resurgence of interest in the 90s corresponding to the development of much less expensive and lower-power receivers than were previously available<sup>40,41</sup> One of the key insights was the use of a range-gate sampler that continually sweeps through a cycling delay. This technique is also used in oscilloscopes to be able to super-sample periodic signals. While most of this initial research focused on RADAR applications, in the 1990s more research shifted into UWB as a short-range, high-bandwidth communication channel intended to stem the proliferation of wires connecting consumer devices like monitors and other computing peripherals.<sup>42</sup>

Allowing the frequency spectrum of a signal to be wide means that in the time domain the signal can operate with very short pulses with sharp transitions. This property is what makes UWB particularly suitable for measuring the time-of-flight of RF pulses, even in the presence of reflections (off of walls, floors, and objects in the area). Consider a signal that is longer than the delay time of its reflections. Though the receiver gets the direct signal first, the reflections overlap the direct signal and make decoding the transmission more difficult, particularly in the presence of noise. By increasing the bandwidth of our transmitted signal we can shorten the duration to be shorter than the reflection delay, so the complete signal can be decoded and any delayed identical packets can be ignored. Note that reflected signals can still be a source of error in cases where the direct signal is blocked by an obstacle, known as non-line-of-site (NLOS) conditions. In these cases the receiver can mistake a reflected signal for the direct, which over-estimates the range.

We use what is known as *Two-Way Ranging* (TWR) to determine the distance between our device (the *Tag*) and several fixed *Anchors* that are at known locations. Without noise we could then solve for

<sup>38</sup> S. Gezici et al. "Localization via ultra-wideband radios: a look at positioning aspects for future sensor networks". 4 (July 2005).

<sup>39</sup> Gerald F Ross. "Transmission and reception system for generating and receiving base-band pulse duration pulse signals without distortion for short base-band communication system". US3728632 A. Apr. 1973.

<sup>40</sup> T. McEwan and S. Azevedo. "Microwave impulse radar" (1996).

<sup>41</sup> Thomas E. McEwan. "Ultra-wideband receiver". US5345471 A. Sept. 1994.

<sup>42</sup> M.Z. Win et al. "History and Applications of UWB [Scanning the Issue]". 2 (Feb. 2009).

the location of the tag analytically using trilateration. Given that noisy signals are inevitable however, there is often no analytical solution to the trilateration problem, so we have implemented the particle filter described in the *Particle Server* chapter.

### Two-Way Ranging

In a two-way ranging configuration, two devices exchange messages back and forth to determine the round-trip time-of-flight between them, from which they determine the distance. Specifically we use a variant known as Symmetric Double-Sided Two-Way Ranging (SDS-TWR) which can compensate for clock drift between the two devices.<sup>43</sup>

A ranging exchange is shown in Figure 6, where  $\tau_r$  is a round-trip time and  $\tau_p$  is the device processing time. First it's important to note that the DecaWave API gives the ability to schedule an outgoing message for some time in the future with the same accuracy that is used for message timestamping, so the device's processing time includes both the time spent actually processing the message and a delay. This allows the processing time to be kept consistent even if the firmware takes a varying amount of time to handle the message. At first, it seems that the time of flight  $\tau_f$  could be determined by the tag using eq. 6.

$$\tau_f = \frac{1}{2}(\tau_{r1} - \tau_{p1}) \quad (6)$$

Unfortunately this neglects the real-world fact that the clocks on the two devices are in general not running at exactly the same speed. To resolve this we introduce  $\epsilon_a$  and  $\epsilon_t$  as the clock speed error on the anchor and tag. We'll define  $\tau_n^{[a]} = (1 + \epsilon_a)\tau_n$  as the time period  $\tau_n$  as measured by the anchor, and similarly for the tag.

So while eq. 6 gives the correct answer in theory, we can only measure time relative to the local clock, so what we get is

$$\begin{aligned} \hat{\tau}_f &= \frac{1}{2}(\tau_{r1}^{[t]} - \tau_{p1}^{[a]}) \\ &= \frac{1}{2}((1 + \epsilon_t)\tau_{r1} - (1 + \epsilon_a)\tau_{p1}) \end{aligned}$$

so the error  $\hat{\tau}_f - \tau_f$  is given by

$$\begin{aligned} &\frac{1}{2}((1 + \epsilon_t)\tau_{r1} - (1 + \epsilon_a)\tau_{p1} - \tau_{r1} + \tau_{p1}) \\ &\frac{1}{2}(\epsilon_t\tau_{r1} - \epsilon_a\tau_{p1}) \end{aligned}$$

<sup>43</sup> Sources of Error in DW1000 Based Two-Way Ranging (TWR) Schemes. Application Note APS011. 2014.

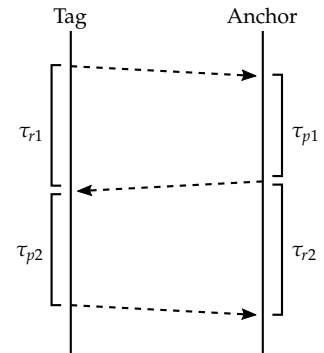


Figure 6: A Symmetric Double-Sided Two-Way Ranging exchange between the tag and an anchor

Because the time of flight is typically at least several orders of magnitude smaller than the processing time, we can simplify by assuming  $\tau_{r1} \approx \tau_{p1}$ , (that is, the overall ranging time is dominated by the receiver's processing time) so the error is given by

$$\frac{1}{2}(\epsilon_t - \epsilon_a)\tau_{p1}$$

So the error is proportional to the processing time and the difference in error. For real world clock drifts and processing times this error can be significant, so we use the full exchange shown in Figure 6. Now the time of flight should be  $\frac{1}{4}(\tau_{r1} - \tau_{p1} + \tau_{r2} - \tau_{p2})$ . Following a similar logic, the error is

$$\begin{aligned} & \frac{1}{4} \left( \tau_{r1}^{[t]} - \tau_{p1}^{[a]} + \tau_{r2}^{[a]} - \tau_{p2}^{[t]} - \tau_{r1} + \tau_{p1} - \tau_{r2} + \tau_{p2} \right) \\ & \frac{1}{4} \left( (1 + \epsilon_t)(\tau_{r1} - \tau_{p2}) + (1 + \epsilon_a)(\tau_{r2} - \tau_{p1}) - \tau_{r1} + \tau_{p1} - \tau_{r2} + \tau_{p2} \right) \\ & \frac{1}{4} (\epsilon_t \tau_{r1} - \epsilon_t \tau_{p2} + \epsilon_a \tau_{r2} - \epsilon_a \tau_{p1}) \end{aligned}$$

Making the same simplification as before we get

$$\begin{aligned} & \frac{1}{4} (\epsilon_t \tau_{p1} - \epsilon_t \tau_{p2} + \epsilon_a \tau_{p2} - \epsilon_a \tau_{p1}) \\ & \frac{1}{4} (\epsilon_t - \epsilon_a) (\tau_{p1} - \tau_{p2}) \end{aligned}$$

So the error is still proportional to the *difference* in clock errors, but now is proportional to the difference in processing time on both sides, instead of the absolute processing time.

While this approach works with for a single tag and small number of anchors, each ranging measurement takes four messages (three for the ranging and one for the anchor to report back the calculated range), and ranging to each anchor must be done sequentially, which adds error if the tag is in motion. Future work will implement a Time-Difference-of-Arrival (TDOA) approach which will only require a single outgoing message from the tag that will be received by all anchors within communication range. This approach shifts complexity to the infrastructure, which must maintain time synchronization between all the anchors. Traditionally UWB systems have used wired synchronization cables, but recent work<sup>44</sup> in wireless clock synchronization has shown that at least in controlled lab settings it is possible to get sufficient clock synchronization to accurately locate tags without wiring the anchors together.

<sup>44</sup> Carter McElroy, Dries Neiryck, and Michael McLaughlin. "Comparison of wireless clock synchronization algorithms for indoor location systems". 2014.

# *Bluetooth Low-Energy*

The HearThere head tracking PCB communicates with the user's mobile phone or computer over Bluetooth Low-Energy (BLE), also known as Bluetooth Smart. For more details on the BLE protocol, "Getting Started with Bluetooth Low Energy"<sup>45</sup> is an excellent and concise starting point.

<sup>45</sup> Kevin Townsend et al. *Getting Started with Bluetooth Low Energy: Tools and Techniques for Low-Power Networking*. 1 edition. Apr. 2014.

## *Connection Interval*

A BLE Connection is a contract between two devices, wherein the devices agree to establish radio contact at a specified interval (the *Connection Interval*) and on a predefined sequence of channels (frequency hopping). The connection interval is an important part of a device's design because it determines the latency, jitter, and bandwidth of communications. Because messages are only exchanged during connection events, messages generated during the interval will be delayed until the next event. Bandwidth is affected because devices have a limited buffer to store pending messages, which limits the number of messages that can be sent during a given connection event. In practice this limit is often as low as two to four messages.

Different devices have widely-varying requirements for their connection interval, depending mostly on trading battery consumption for latency and bandwidth. To manage these diverse requirements, when a new connection is being established there is a process of negotiation. The BLE *Peripheral* is able to request a particular range of connection parameters (including the connection interval), but it is up to the *Central* to decide what the final parameters will actually be. The peripheral may make these requests when the connection is first established or at any time in the future. The BLE specification allows connection intervals to be between 7.5 ms and 4 s, but a Central does not have to support the whole range.

Apple's document "Bluetooth Accessory Design Guidelines for Apple Products" says:

The connection parameter request may be rejected if it does not comply with all of these rules:

- $\text{Interval Max} * (\text{Slave Latency} + 1) \leq 2s$
- $\text{Interval Min} \geq 20ms$
- $\text{Interval Min} + 20ms \leq \text{Interval Max Slave Latency} \leq 4s$
- $\text{connSupervisionTimeout} \leq 6s$
- $\text{Interval Max} * (\text{Slave Latency} + 1) * 3 < \text{connSupervisionTimeout}$

Through trial-and-error we have determined that as of January 14, 2015, the lowest connection interval possible on an iPhone 5S is 15 ms to 17 ms, achieved through requesting an interval between 10 ms to 20 ms. Additionally this request must be made after the initial connection negotiation, even if the same parameters were used for the original connection.

## Services

BLE functionality is advertised and provided to a *client* by a *server* through *Services*. To that end, the HearThere head tracker provides two services: The *Attitude and Heading Reference System Service* (AHRS) which provides quaternion orientation estimates and the *Ranging Service* which provides ranges to anchors.

The Ranging Service provides a single characteristic that it uses to notify the client of newly-available range data. The characteristic data format is defined by the struct:

```
typedef struct __attribute__((__packed__)) {
    uint8_t seqNum;
    float rangeMeters[4];
} ble_ranging_range_value_t;
```

Where the `seqNum` field is incremented with each new value and `rangeMeters` is an array of ranges to the 4 anchors in meters. Each value is transmitted in little-endian order. When there is a communication failure to a given anchor during the ranging process, the device reports `-1`.

The AHRS Service contains an Orientation characteristic and a Calibration characteristic. The Calibration Characteristic is a single byte that can be written by the client to put the device in calibration mode, where it will transmit raw IMU data rather than the orientation determined by the sensor fusion algorithm. The Orientation characteristic has the format:



```

typedef enum {
    // used to send the fused data
    AHRS_DATA_QUATERNION,
    // used to send the raw 16-bit integer sensor data
    AHRS_DATA_RAW_DATA
} ble_ahrs_data_type_t;

typedef struct __attribute__((__packed__)) {
    uint8_t sequence;
    ble_ahrs_data_type_t type;
    union {
        struct __attribute__((__packed__)) {
            float w;
            float x;
            float y;
            float z;
        } quat;
        struct __attribute__((__packed__)) {
            int16_t accel_x;
            int16_t accel_y;
            int16_t accel_z;
            int16_t gyro_x;
            int16_t gyro_y;
            int16_t gyro_z;
            int16_t mag_x;
            int16_t mag_y;
            int16_t mag_z;
        } raw;
    };
} ble_ahrs_orientation_value_t;

```

In calibration mode the device simply sends the data as measured, and in normal mode it sends the 4-component quaternion orientation given by the sensor fusion algorithm (see the *Orientation Tracking* chapter).

## *iOS Application*

The HearThere iOS application is built using the Unity3D game engine and written in C#. Though the main user-facing app is intended to focus on sound, we have implemented several features to display various system metrics. The main responsibilities of the application are:

- Managing the BLE connection and receiving data
- Updating the orientation estimate
- Updating the location estimate
- Synchronizing the in-game listener to the real-world user's head
- Placing the virtual audio sources in the game world
- Binaural rendering of the virtual audio sources relative to the listener
- Displaying internal data in the user interface
- Transmitting raw and processed data over OpenSoundControl (OSC)
- Displaying the user's position on a map

Figure 7 shows the first screen the user sees after connecting to the head tracker hardware, which happens automatically on application launch. This view visualizes the software's estimate of the head orientation and allows the user to re-zero the representation to compensate for the lack of an absolute direction reference when the magnetometer is disabled. The left and right navigation buttons to go to the other views.

As the head tracker sends range data to the iOS application, the app collects various statistics and displays them on the screen shown in Figure 8. This also allows the user to set the destination for the OSC messages, where it will send orientation and range data. We currently use the OSC stream for data logging, but it is also useful as an integration point with other systems.

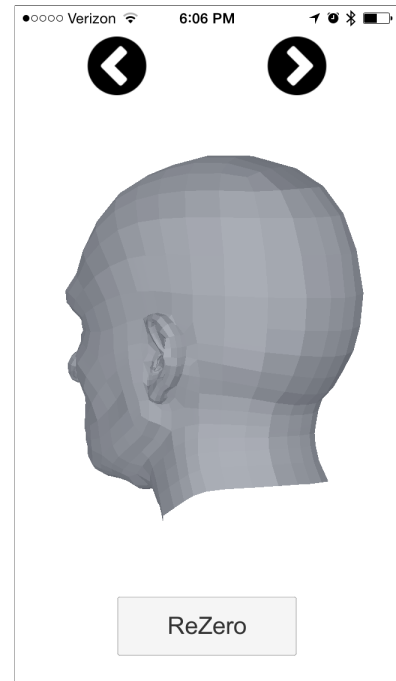


Figure 7: Orientation display in iOS application, which visualizes the orientation estimate by rotating the displayed head

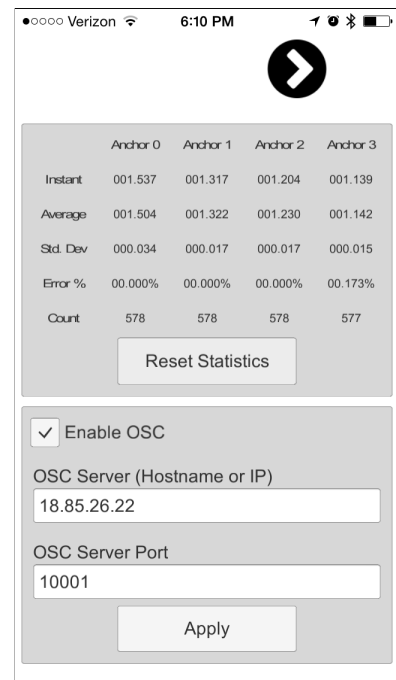
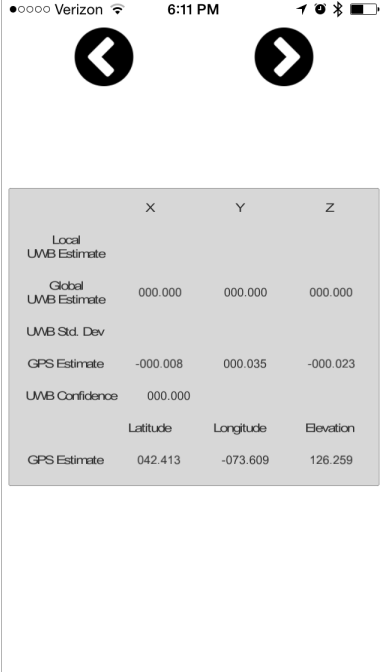


Figure 8: Range statistics display, which reports the raw data as well as accumulated statistics for each anchor

The UWB location estimate and standard deviation, GPS fix, and fused location estimate are displayed in Figure 9, as well as updating the map view in Figure 10. The Local UWB estimate is the coordinates within the UWB Zone coordinate system, and Global UWB is in the Global UWB estimate is within the global Unity coordinate system. The GPS estimate is displayed both in native latitude/longitude/elevation and mapped into the global Unity frame. The UWB Confidence value goes from zero to one and determines the mix of GPS and UWB location used in the fusion.

### *Audio Engine*

While Unity provides a sophisticated authoring environment for placing sonic objects in our world, the built-in spatialization is very basic. It only models interaural level difference (ILD) and optionally a simple lowpass filter to approximate occlusion effects for sources behind the listener. With the recent resurgence of virtual reality there are a number of more sophisticated spatial audio engines now available. We are using the 3DCeption plugin from Two Big Ears<sup>46</sup> which uses a generalized head-related transfer function (HRTF) which captures interaural level and time differences, as well as spectral cues. They also implement a simple shoebox model to generate physically-plausible first-order reflections from the environment.



	X	Y	Z
Local UWB Estimate			
Global UWB Estimate	000.000	000.000	000.000
UWB Std. Dev			
GPS Estimate	-000.008	000.035	-000.023
UWB Confidence	000.000		
	Latitude	Longitude	Elevation
GPS Estimate	042.413	-073.609	126.259

Figure 9: Location statistics screen, which reports the location information from the GPS and UWB systems, as well as the fused location estimate

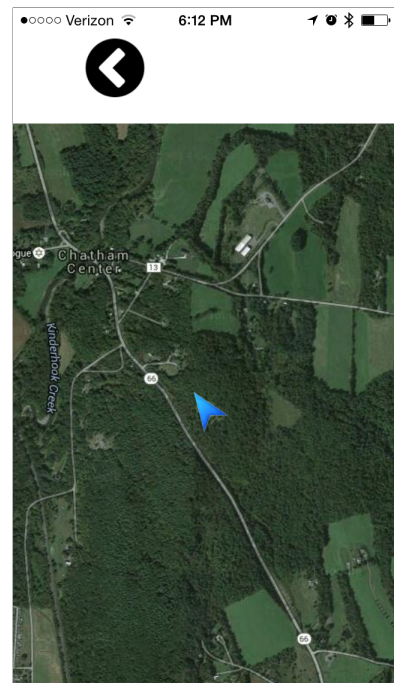


Figure 10: iOS Application Map screen. The map tiles are downloaded on-demand from Google Maps

<sup>46</sup> <https://twobigears.com/>

# Particle Server

In principle we should be able to use ranging data from a number of different fixed locations and analytically solve for our tag location. This process is known as multilateration. In the presence of noise this problem becomes much more difficult, although many approaches are viable. We chose to solve the problem using a particle filter,<sup>47</sup> which has been shown to perform well specifically in UWB-based localization systems.<sup>48</sup> Particle filters are particularly attractive because they provide a straightforward way to use measurable statistics (such as the variance of ranging data) to create a likelihood model that can generate a complex probability distribution.

We implemented our particle filter system<sup>49</sup> in the Julia programming language and added an HTTP interface using the Morsel.jl framework. Clients can make an HTTP request to the root of the web server, which initializes a fresh set of particles and sends a response to the client with a link that they can use to submit sensor updates (see Figure 11).

We initialize our particle system to uniformly fill the capture volume with 1000 particles. The state vector simply represents the estimated  $x$ ,  $y$ , and  $z$  coordinates. Each time the server receives a new set of range data it updates the model by:

1. Stepping the particles with the process model. Because our state does not model any dynamic characteristics such as velocity, our process model is simply a gaussian-distributed random walk that models the space of possible movements since the last update.
2. Setting the weight of each particle based on the likelihood given the sensor data. Our likelihood function models the range data as normally distributed about the true value, using the variance from our calibration measurements.
3. Resampling the data using the weighted particle cloud as a proxy for our estimated probability distribution. To maintain

<sup>47</sup> Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic Robotics*. Cambridge, Mass, Aug. 2005.

<sup>48</sup> J. González et al. "Mobile robot localization based on Ultra-Wide-Band ranging: A particle filter approach". 5 (May 2009).

<sup>49</sup> <https://github.com:ssfrr/ParticleFilters.jl>

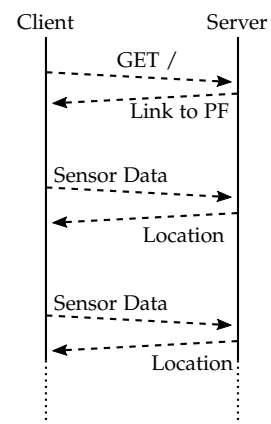


Figure 11: Initializing the particle server over HTTP and updating with new sensor data

particle diversity we employ the low-variance resampling technique described in Thrun et. al.

4. Returning to the client the mean and standard deviation vectors of the particle cloud

In step 2 we evaluate the likelihood function  $\mathcal{L}(x; A, z)$  at each particle location  $x$  in the state space (here just the estimated  $x$ ,  $y$ , and  $z$  coordinates) given the set of anchor locations  $A = \{a_0, a_1, a_2, a_3\}$  and the observation vector  $z$ , which is the measured ranges to the 4 anchors.

$$\mathcal{L}(x; A, z) = \begin{cases} 0, & 0 > y > 2.6 \\ \prod_{i=0}^3 \left( \alpha + (1 - \alpha) \mathcal{N}(\|x - a_i\|; z_i, \sigma^2) \right), & \text{otherwise} \end{cases}$$

where  $\mathcal{N}(\|x - a_i\|; z_i, \sigma^2)$  evaluates the likelihood of the state vector  $x$  given a single range measurement, by treating the range as normally distributed. In our case  $\sigma^2$  is a constant based on our calibration measurements. We use the constant  $\alpha$  (0.2 for our experiment) so that there is still a nonzero probability outside of our gaussian window.

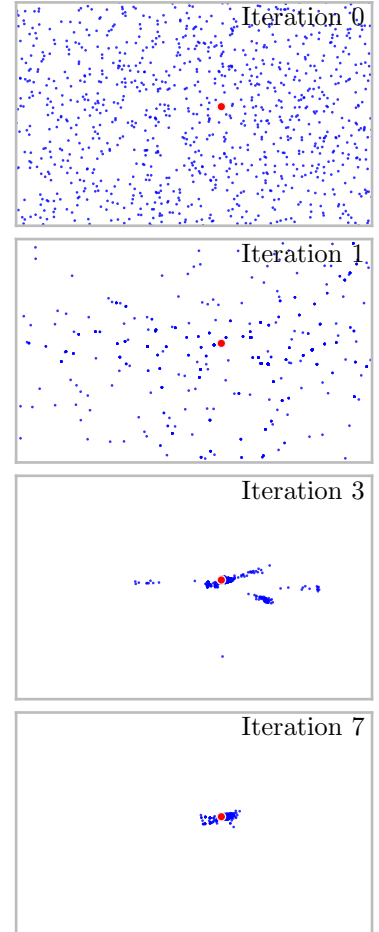


Figure 12: Four iterations of the Particle Filter, projected onto the  $xz$  plane. The true location is shown in red.

## *GPS Integration*

While the Ultra-WideBand radio provides much more accurate location than GPS, the fact remains that much of the time the user will not be near any installed UWB infrastructure. In these cases we want to be able to fall back to GPS. Also we want the user to be able to move between non-contiguous UWB zones which operate in local coordinate systems, and virtual audio sources need to be describable in terms of both global coordinates (latitude, longitude, and elevation) and local coordinates ( $x, y, z$  in meters). Though our current hardware limits us to a single UWB zone with four anchors, in the near future we plan on expanding coverage.

To satisfy these constraints we have implemented in Unity a multi-scale location representation. Any object in Unity can be tagged with geographic coordinates. When the application starts, the user is placed at the origin of the game world, and all geo-tagged objects are placed around them using their coordinates relative to the user's (from the phone's GPS). This placement uses the Mercator projection, which preserves angles and shapes in the local vicinity of the user, and is also simplifies retrieving map image tiles, as it is used by the main map tile providers. Each UWB zone is also a geo-tagged Unity object, though instead of being a sound-producing object itself, it simply serves as a parent object (known as an *empty*) for all the virtual sound sources within that zone. Objects within the zone can use local coordinates relative to the zone origin, and the zone parent handles scaling, rotating, and placing that local coordinate system into the global Unity coordinate system

This allows a sound designer to work at both scales easily. They can tag large objects such as cities, buildings or landmarks with geographic coordinates, and can easily work with local cartesian coordinates relative to the UWB zone origin when working at smaller scales.

The iOS application has access to the raw UWB range information, as well as whether the device is successfully communicating with the anchors at all. In the case that we don't have any range

information, we are clearly outside of the UWB Zone and fall back to GPS exclusively. As we approach a UWB zone, the device starts to collect range information and sends it to the Particle Server to compute a location estimate (in the local UWB coordinate system). Along with the location estimate the server also reports the standard deviation of the particle cloud. Given good ranging data that has a valid solution, the particle cloud tends to converge in the neighborhood of the solution with a low variance, so we use this to determine when we can rely on the UWB ranging data instead of GPS. Currently the particle filter is used exclusively for the UWB data and fusing the UWB estimate with the GPS estimate is a separate step. In the future the GPS estimate could be integrated directly into the likelihood estimate of the particle filter.

# Calibration

When using any kind of sensor, calibration is critical. With that in mind we took care to characterize and calibrate the UWB ranging, accelerometer gain and bias, gyroscope gain and bias, and magnetometer gain and bias.

## UWB Range Calibration

Decawave provides a application note<sup>50</sup> that details the various sources of error in a UWB ranging system and how to mitigate their effects. One of the most important necessary calibrations is the antenna delay, which accounts for the extra time it takes for the signal to get from the output pins of the chip through the antenna and into the air (and the inverse on the receiving side). Our initial ranging tests were between the HearThere head tracking board and the DecaWave EVB1000 evaluation board.

We started with the DecaWave-recommended value of 16436 ticks, and took measurements at three known distances (50 cm, 150 cm and 250 cm) at several delay values (16436, 20000, 21250, 22500, 25000). The delay values were determined iteratively as we tested. We read the average values displayed by the EVB1000 (as seen in Figure 13) in our measurements.

Delay (ticks)	50 cm	150 cm	250 cm
16436	1967 cm	2071 cm	2175 cm
20000	720 cm	825 cm	927 cm
21250	260 cm	369 cm	464 cm
21839	54 cm	159 cm	254 cm
22500	N/A	N/A	13 cm
25000	N/A	N/A	N/A

The final value of 21839 was determined by running a linear regression on the data from the three closest values (20000, 21250, and 22500).

With the antenna delay programmed into the firmware, we also

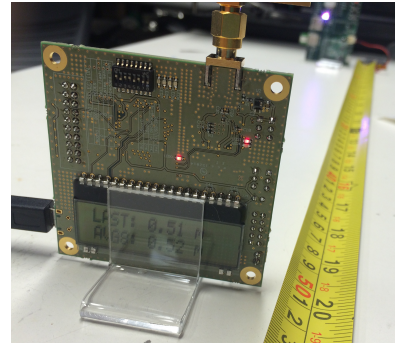


Figure 13: Decawave EVB1000 with display screen used in antenna delay calibration

<sup>50</sup> Sources of Error in DW1000 Based Two-Way Ranging (TWR) Schemes.

Table 2: Reported ranges vs. actual, for different antenna delay settings

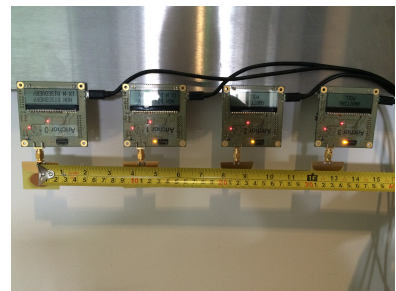
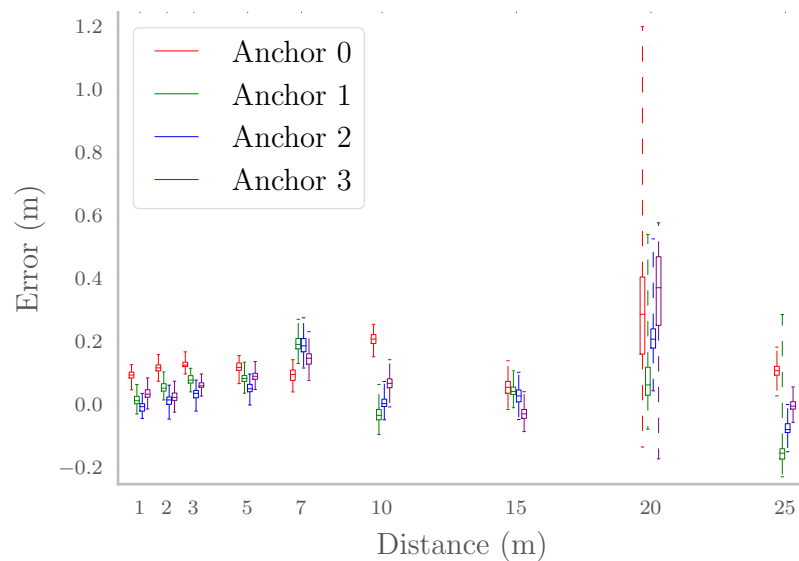


Figure 14: Anchor configuration for range calibration



tested ranging to all four anchors. We mounted them to the wall as shown in Figure 14 and measured data at distances from 1m to 20m. The results are plotted in Figure 15. It does not appear that the measurement variance is very sensitive to distance (this matches previous work<sup>51</sup>), but it is unclear from this data whether the biases are distance-dependent or due to some unobserved variable. The results measured at 20 m are somewhat mysterious, but might be due to multipath effects or interference. We averaged the error for each anchor using the ranges 1 m, 2 m, 3 m and 5 m and stored it in the iOS application as an offset to be applied to each range measurement.



<sup>51</sup> Chiu and O’Keefe, “Seamless outdoor-to-indoor pedestrian navigation using GPS and UWB”.

Figure 15: Error data from calibration of all four anchors

### IMU Calibration

Each sensor in the IMU required a different approach to calibration. For the gyroscope we first measured while the device was stationary. We then mounted it on a turntable (see Figure 17) and measured the reported values at both 33RPM and 45RPM, for each of the three axes. The bias and gain calibration values were determined by linear regression for each axis.

The magnetometer and accelerometer calibration were both done by recording data while rotating the device randomly, trying to get good coverage of the whole sphere. Bias and gain were calculated using the minimum and maximum reported values for each axis. This calibration could be improved in the future by fitting an ellipse to the recorded data, but in practice the MEMS

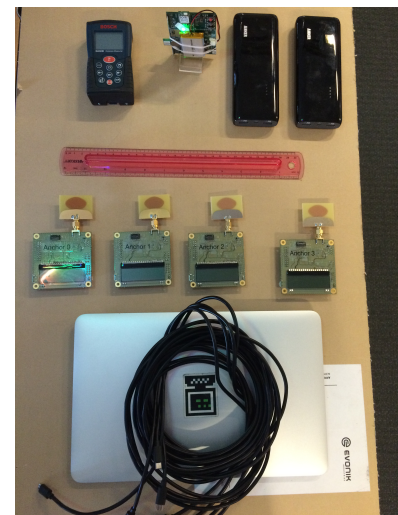


Figure 16: Equipment for anchor calibration

accelerometer seems to be accurate enough so as not to require calibration (our measured bias was very close to zero and gain very close to 1 for all axes). For the magnetometer the most important calibration is the bias due to manufacturing variance and hard iron effects. Magnetometer gain is somewhat less problematic, though differences in gain between the axes can cause distortions in the measurements.

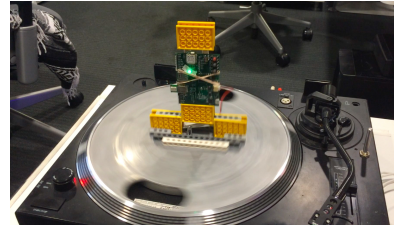


Figure 17: Turntable configuration for calibration gyro gain

# Quantitative Evaluation

## Experimental Setup

To evaluate the accuracy of our tracking system we collected data from the system while the user's head was also instrumented with a optical motion capture markers. We used a six-camera OptiTrack<sup>52</sup> motion-capture system. Figure 18 shows an overhead view of the configuration with the cameras and anchors in a level plane near the ceiling, along with the path from the OptiTrack data projected onto the X-Z plane.

We determined the positions  $(x_2, y_2, z_2)$  and  $(x_3, y_3, z_3)$  of anchors A2 and A3 respectively using the OptiTrack system by placing a tracking marker at their location, as shown in Figure 19. We then used a laser range-finder to measure the distances  $r_{20}$ ,  $r_{21}$ ,  $r_{30}$ , and  $r_{31}$  from the known anchors (A2 and A3) to the unknown (A0 and A1). We could then trilaterate the positions of A0 and A1, constrained to be in the same plane as the known anchors and using the known geometry to disambiguate between the two solutions to the intersecting circles.

We added tracking markers to the user's head as shown in Figure 22, as well as placing the HearThere head tracker PCB under the top elastic strap. With the coordinate systems of the OptiTrack system aligned with that of our tracking system, the user walked around the capture volume while we recorded the location and orientation of the head. During the test we captured the OptiTrack data to a local file and the HearThere iOS application streamed range data using the OpenSoundControl (OSC) protocol to a laptop which recorded the data.

Latency comparisons based on these data were impossible because of the unknown latency of the OptiTrack system and the unknown clock relationship between the different machines used. As such the first steps in processing the data were to re-sample the two datasets (OptiTrack and HearThere range data) to a common sampling rate and time-align the data by maximizing the cross-correlation.

<sup>52</sup> <http://www.optitrack.com/>

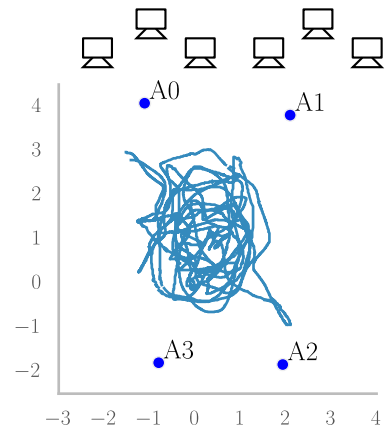


Figure 18: Experimental setup with six OptiTrack cameras and four UWB anchors, with the path walked during the experiment



Figure 19: Localizing A2 using optical tags

### Ranging Accuracy

We compared the raw range data measured over UWB to our expected range (computed from distances between OptiTrack’s reported head location and the anchor locations). The results are shown in Figure 20.

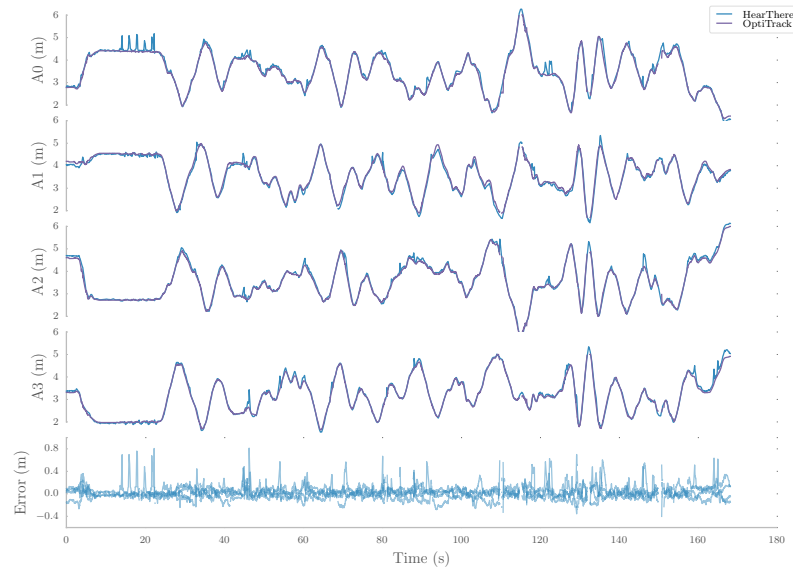


Figure 20: Measured range from the HearThere hardware compared to expected range based on OptiTrack location data

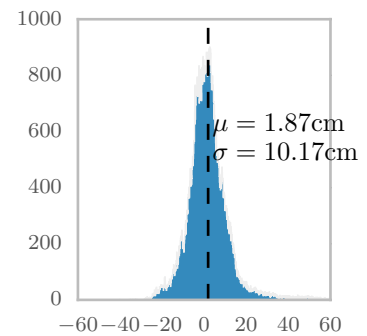


Figure 21: Overall ranging error (in cm)

We see that the HearThere UWB system tracks the expected ranges very well, though there are a few spikes due to head occlusion blocking the direct path, so the UWB is reading NLOS (non-line-of-site) paths. In Figure 21 we see that despite our calibration we have a mean error of 1.87 cm. This is most likely because of errors in calibration or inaccuracies in our anchor location measurements. The standard deviation of our range error is 10.17 cm.

### Localization Accuracy

To evaluate HearThere’s localization accuracy we used the measured range data and ran it through our particle filter implementation. Though these measurements were done offline, the data was processed sequentially using the same algorithm that our particle filter server uses, so the performance here should be representative of the algorithm running in real time. Runtime of the algorithm is substantially faster than real-time (on a 2011 MacBook Air), so compute performance is not an issue.



Figure 22: Optical markers placed on the user’s head for tracking

Figure 23 shows the position measured by the OptiTrack system compared against 20 runs of the particle filter on our measured range data, to indicate the variance introduced by the nondeterministic algorithm. From the plot it is clear that while the filter is capable of tracking the location most of the time, it consistently loses track during some portions of the test, at which point it becomes very inaccurate. This points to a clear path to improvement which is identifying the cause of these errors. Figure 24 shows the error histograms under two conditions. The *Unfiltered* error is simply the difference between the value reported by the particle filter and the value reported by the OptiTrack system. To represent the behavior during the times the algorithm is tracking well, we also analyzed the error for the region from approximately 20 s to 100 s (the *Filtered* error). This removes the long error tails in the error distribution which drive the standard deviations up.

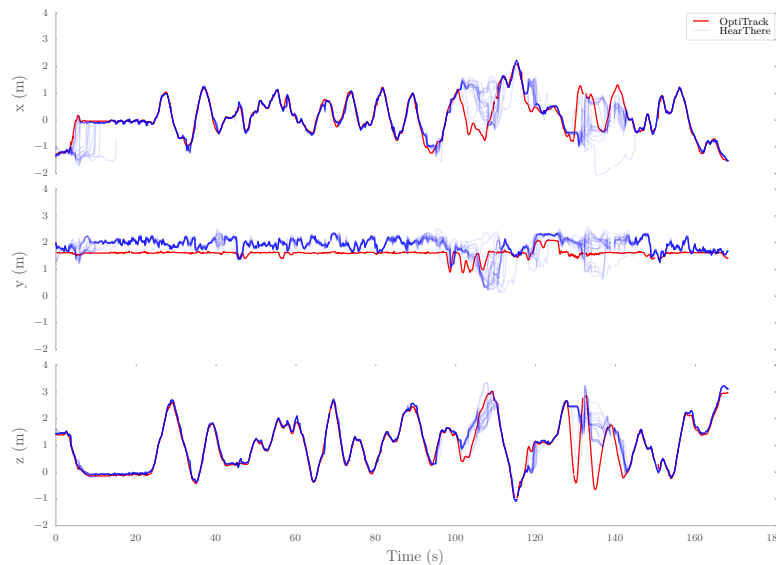


Figure 23: Location from OptiTrack compared to 20 runs of the particle filter fusion algorithm on the HearThere ranging data

## Orientation

To evaluate the rotational accuracy of the HearThere system we again compared to the OptiTrack installation with the same configuration as in Figure 18. Figure 25 shows the overall movement made, plotted both for the OptiTrack system and HearThere. For several orientations the OptiTrack system lost its lock on our head marker, as indicated by gaps in the plotted data. The performance of the OptiTrack system could be improved with a more robust

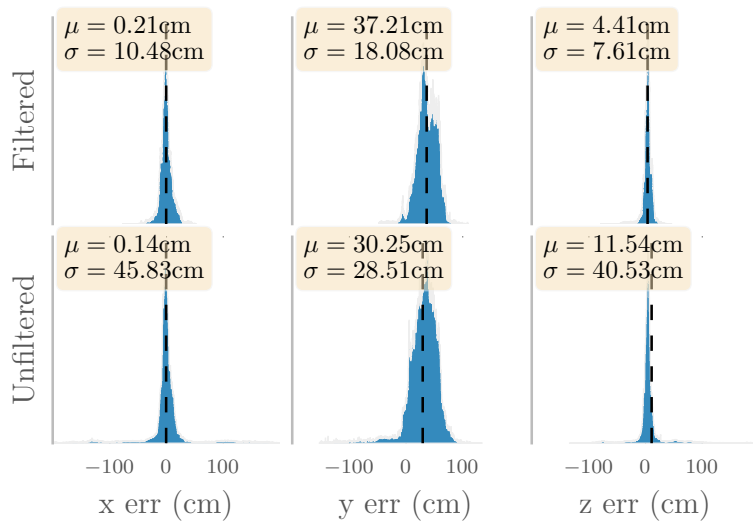


Figure 24: Error histogram of  $xyz$  data from the particle filter. The filtered data is only from the timespans where the algorithm was tracking, the unfiltered includes the times when the algorithm totally loses track. Notice that the long tails of the unfiltered data cause it to be strongly non-gaussian and drives the standard deviation up.

camera configuration.

The rotations have been phase-unwrapped to illustrate the full rotations and remove distracting discontinuities. As a result the scale of the plots is dominated by the several full rotations, so Figure 26 shows only the beginning and end of the test, to show more clearly the accuracy of the system. Both plots are compared to the quaternion error (using the arclength distance metric<sup>53</sup>), measured in radians.

It may be more intuitive to investigate the yaw/pitch/roll errors separately, as in Figure 27. This reveals the cyclic nature of the measured error (though also introduces artifacts near the singularity points of pitch =  $\pm \frac{\pi}{2}$ ). The dependence of the error on rotation points to a possible source of error, which is misalignment between the HearThere’s concept of the global frame and the actual frame. Both the HearThere and OptiTrack provide a mechanism to *re-zero* their tracking so that the current orientation becomes the identity quaternion (1 0 0 0), but HearThere is currently only designed to correct for mis-alignment about the  $y$  axis (up) and if the system is tilted during re-zeroing it can cause alignment errors in subsequent rotations.

We also measured the device’s drift both with and without the magnetometer. In Figure 28 we see that there is very little drift in any axis with the magnetometer, and about  $0.3 \frac{\text{mrad}}{\text{s}}$  in the yaw axis without. The linear drift (rather than a noise-driven random walk) indicates that there is a constant gyro bias and the main issue is calibration. Future work will experiment with

<sup>53</sup> Du Q. Huynh. “Metrics for 3D rotations: Comparison and analysis”. 2 (2009).

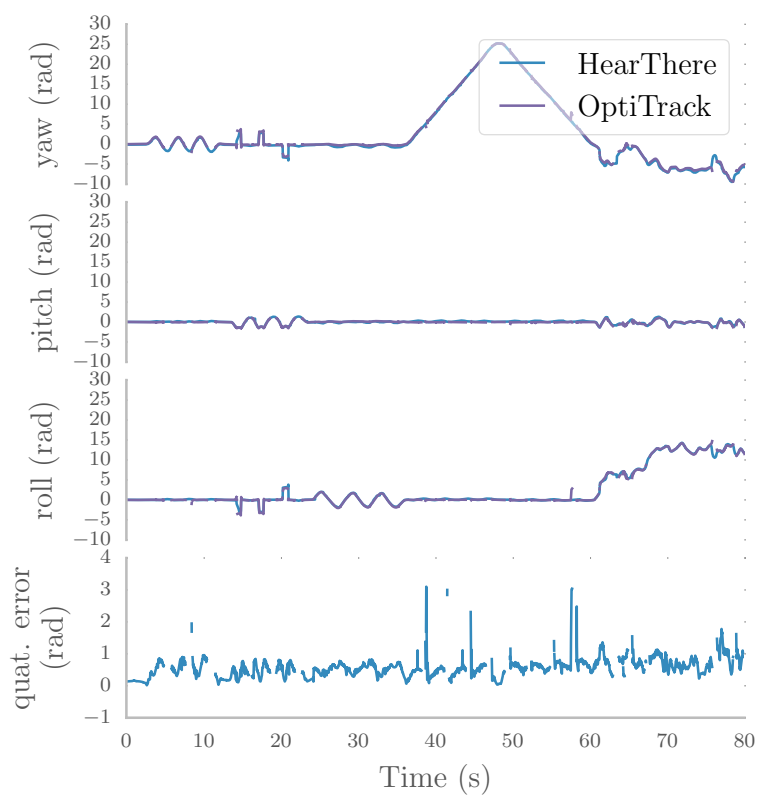


Figure 25: Overall comparison of HearThere and OptiTrack orientation measurements. This test run included separate head oscillations in the yaw, pitch, and roll axes, three full rotations clockwise, four full rotations counter-clockwise, and a period of unstructured random head movements. Error metrics are in radians

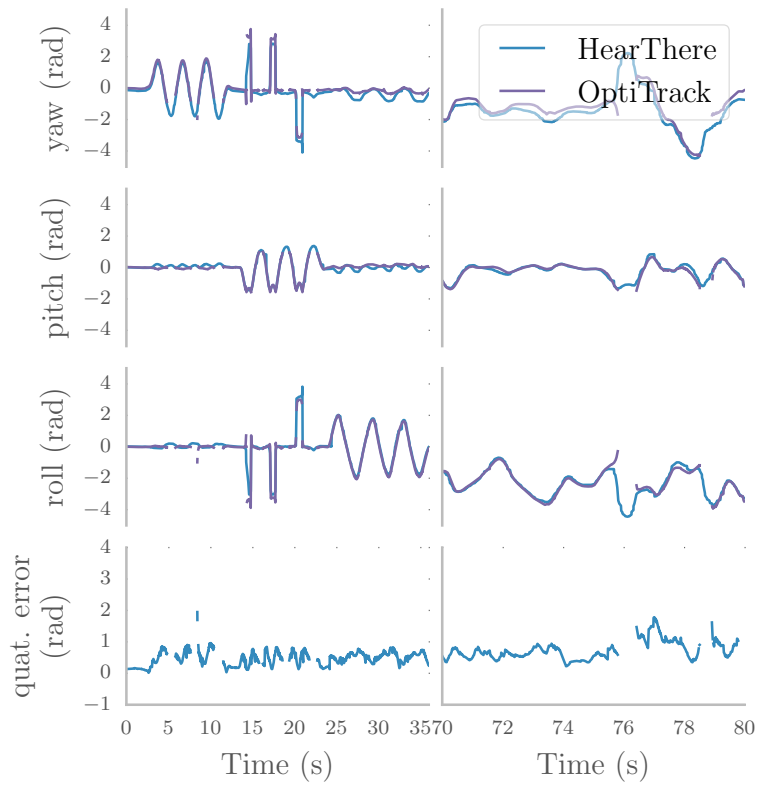


Figure 26: Detail view of the beginning and end of the test period



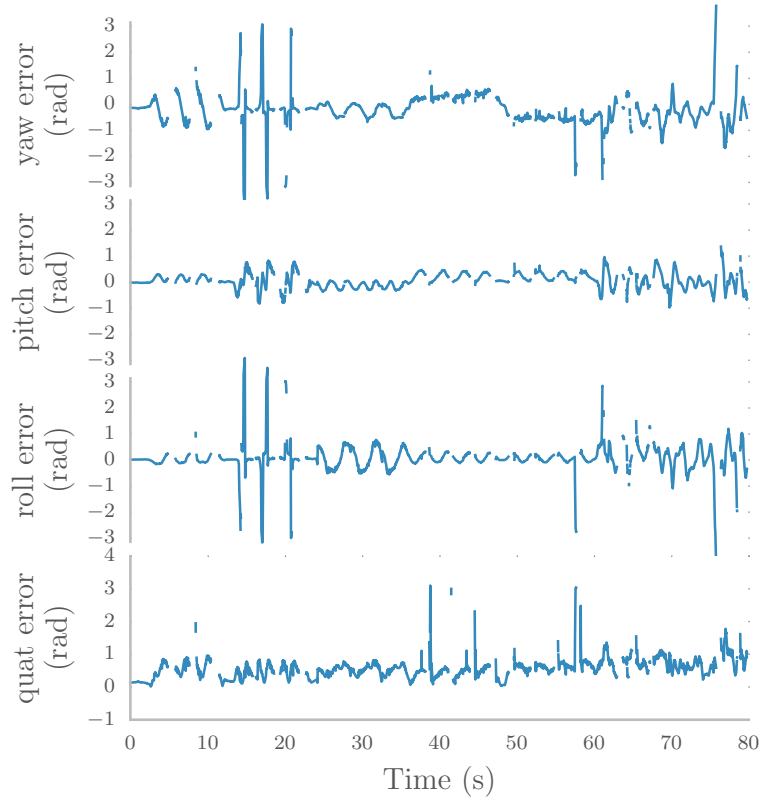


Figure 27: Overall error compared to Yaw/Pitch/Roll components. Note the glitches in the yaw and roll errors caused by singularities when pitch approaches  $\pm \frac{\pi}{2}$

dynamically compensating for the gyro bias, or allowing field calibration. We also expect some improvement by modeling the effects of temperature on the gyroscope bias. Even with improved calibration we still expect some amount of drift without any fixed reference, so we also plan to explore using movement data from the location tracking to help correct the orientation drift, even without the magnetometer.

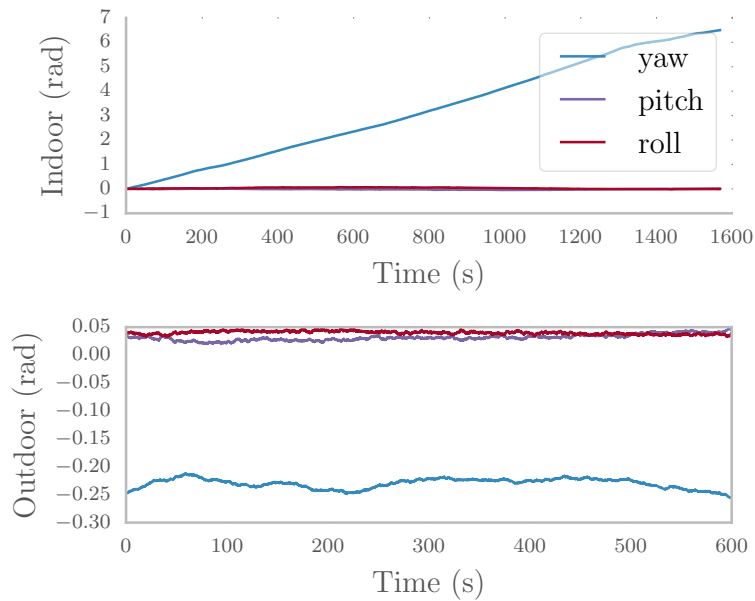


Figure 28: Drift in Outdoor Mode (with magnetometer) and Indoor Mode (without)

## *User Study*

In order to demonstrate end-to-end system functionality, as well as evaluate whether it represents a useful step towards our goal of a seamlessly-integrated auditory augmented system, we conducted a user interface study. The study took place in an outdoor plaza on MIT campus. The UWB anchors were set to cover part of the test area, and outside that zone the system would be forced to rely on GPS for localization. We conducted the study with six volunteers between the ages of 23 and 36, all of whom were students or visiting researchers at the MIT Media Lab. Four were male and two were female.

### *Procedure*

We created eight virtual audio sources scattered within the test area. The user's task was to walk around the test area with the HearThere head tracker and attempt to locate the sources using their ears. The test was conducted in two phases, A and B, with four virtual audio sources each. The samples were the same for each phase, but the locations were different. The subject always did phase A before phase B, but half the participants (LostCobra, FastIbis, and SpacyFrog) were randomly selected to use Etymotic ER-4 in-ear headphones first and then Aftershokz Sportz 3 bone conduction headphones, and the others (StrongCrow, IndigoCheetah, and RockingGoldfish) reversed the order. The sounds were not modified to account for differences in playback equipment (bone conduction vs. in-ear), but the users were able to adjust the volume freely.

The audio sources were a female voice, bird sounds, chickens, and a solo saxophone. The sources were not attached to obvious visible objects or landmarks, but were at intersections of the tiles on the ground. Their locations were recorded ahead of time using a builder's level and laser rangefinder. They were placed so that we expected roughly half of them (0-3) to be within range of the UWB anchors. After reviewing the data it appears that source

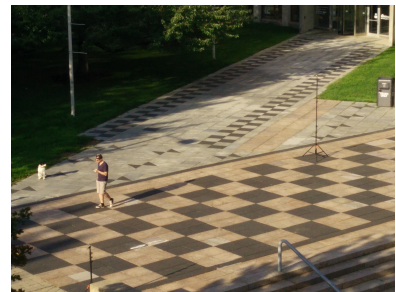


Figure 29: Test area, with user.

4 was often picked up by UWB as well, so in our comparison analyses we consider it to be inside the UWB zone.

Source	Scene	Description	UWB	$x$	$z$
0	A	Chickens	Yes	0.00 m	2.56 m
1	A	Saxophone	Yes	2.56 m	7.50 m
2	B	Birds	Yes	1.90 m	10.42 m
3	B	Female Voice	Yes	4.72 m	5.57 m
4	A	Birds	Yes	3.21 m	-5.67 m
5	B	Chickens	No	-0.23 m	-8.57 m
6	A	Female Voice	No	4.06 m	19.94 m
7	B	Saxophone	No	12.99 m	16.43 m

Table 3: Sources used in the study

During the tests the user placed a label on the ground where they thought the source was located, and after both phases we recorded the distance from the tape to the actual positions. The subjects knew the names of the sounds and in some cases some additional description was given, but they had not heard the sounds prior to the test. The results were marked *NA* either if the user was not able to decide on a location within the 10-minute time limit or if the error distance was greater than 8.5 m.

After the task the user completed a short survey. The full results are listed in Appendix B for reference.

### *Survey Results and Themes*

We expected that the task would be somewhat easier with the in-ear headphones than with bone conduction, as the bone conduction headphones have a more limited frequency response which can interfere with spectral cues, and also play at generally lower volume. Our survey results agree with this hypothesis, with three reporting the task was easier with headphones, one with bone conduction, and two stating no preference. Note though that with such a small sample size this data does not have much statistical power.

Five out of six users reported confusing the real and virtual sounds at some point, particularly the birds. This is a promising signal that the spatialization and externalization are at least somewhat effective, and interesting in that the plausibility of the sources is an important factor (e.g. the subjects knew there was no saxophone player).

When looking for the chickens, I couldn't help but look down as if I was searching a chicken, but for the voice sound I looked

straight forward. some sounds caused an reaction [sic] as if someone appeared suddenly behind me.

There are also common themes that point to areas of improvement. Several subjects mentioned that the sounds occasionally would abruptly jump from one location to another. The jarring nature of these shifts could be mitigated with a slower filter on the location so that changes happen more smoothly, but it also points to tracking discontinuities. Subjects also mentioned that the volume differences between sources and low volume when using the bone conduction headphones made the task more difficult, indicating that more care should be taken to balance these levels in the future. Multiple users also noticed that the volume of the sounds dropped off too steeply with distance. The distance fading effects were exaggerated during this test in an effort to reduce distraction and help the users separate the sources, but this feedback indicates that a more natural rolloff would have been easier.

### *Subject Localization Error*

Figure 30 shows the error for each user and for each source.

During phase A (sources 0, 1, 4, and 6) the user StrongCrow thought that the source location was limited to poles in the area, which contributed to their relatively poor performance. These results have been withheld from the following analysis. For both phases of RockingGoldfish's test the application was not connected to the particle filter server, so the run used only GPS.

Figures 31 and 32 show the error data grouped by the scene (Scene A was always presented before Scene B) as well as the headphone type. While the experimental design doesn't support very strong head-to-head conclusions we can qualitatively note a few things about this data. All four of the failures occurred with the bone conduction headphones. In three of those cases the stickers were not placed at all because the user was unable to hear the source. This is likely because the volume was much less with the bone conduction headphones, which was exacerbated by the overly-steep volume roll-off with distance. Despite these challenges subjects were clearly still able to localize the sources relatively accurately, and we expect performance would improve with sounds more tailored to bone conduction.

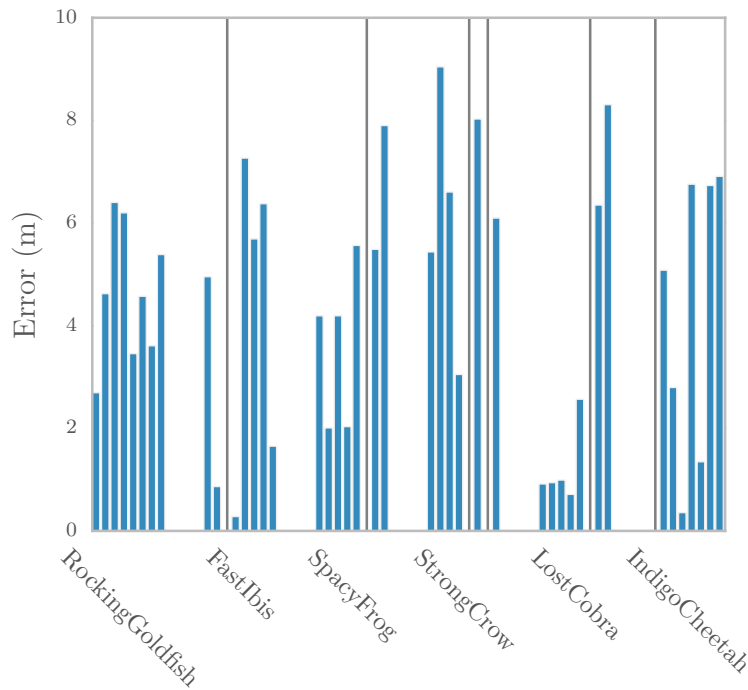


Figure 30: Localization error. Sources are shown in order (0-7), and failures are represented by vertical lines

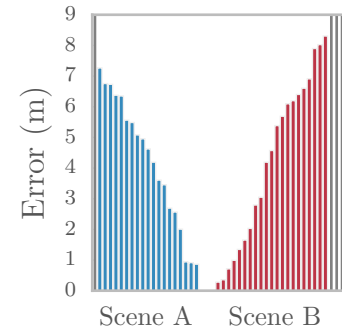


Figure 31: Localization error grouped by scene

One of the most striking contrasts is the difference between sources with and without UWB coverage, as seen in Figure 33. This supports the conclusion that users were able to use the higher-precision and lower-latency localization to get a more accurate idea of where the sources were.

### Tracking Results

Figure 34 shows the accumulated tracking data for all users. The area covered by the UWB tracking system was larger than we expected, and only sources six and seven are consistently outside of the UWB zone, with five on the fringe. The resolution of the GPS tracking is also clearly visible as a grid.

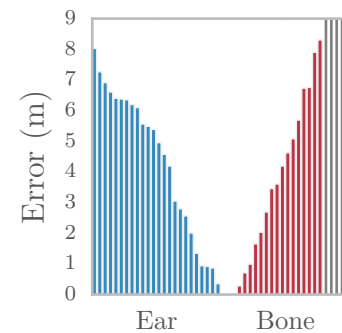


Figure 32: Localization error grouped by headphone type

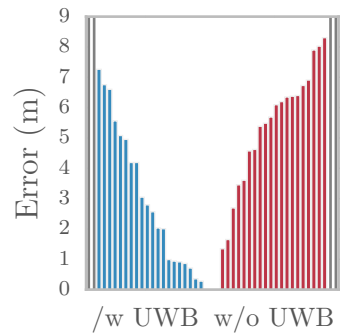


Figure 33: Localization error grouped by whether the source was covered by UWB or not

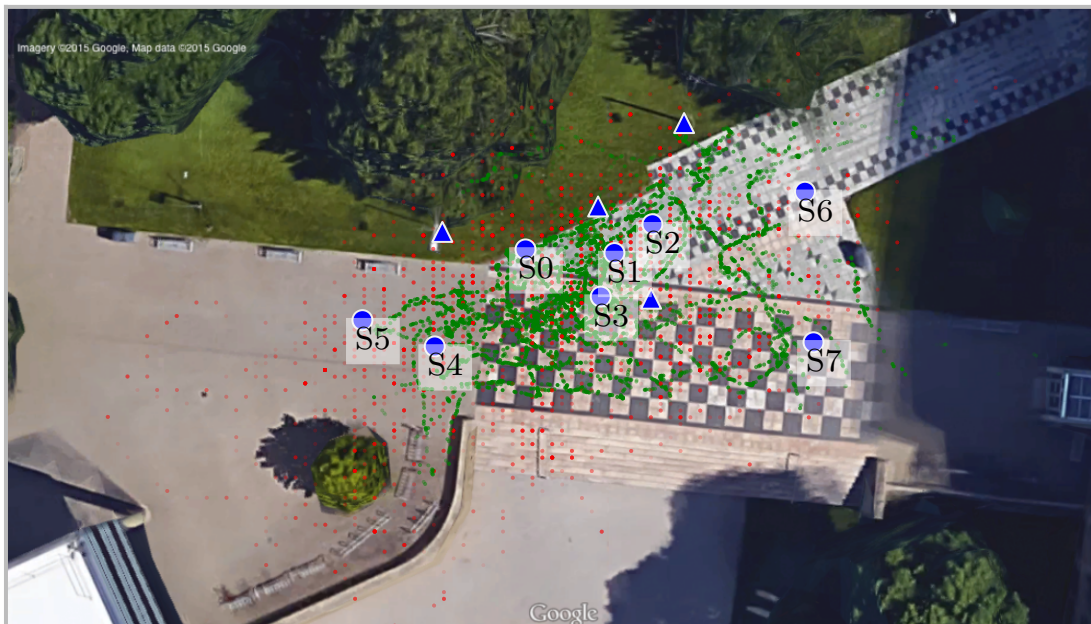


Figure 34: Tracking data accumulated for all users during the experiment. Triangles are UWB Anchors, Circles are audio sources. The green trace is the output of the particle filter with the opacity representing the confidence. The red trace is the GPS estimate

## *Conclusions and Next Steps*

In this work we have demonstrated the the HearThere system can deliver accurate orientation and location estimates to a mobile phone in real time. We described our implementation fusing GPS location with the Ultra-WideBand-based location at runtime, as well as a framework for sound designers to create content at both large and small scales. We have also described the problems, solutions, and design decisions involved with building a fully custom ultra-wideband localization system and orientation tracker. As this technology becomes more widely available and more developers and researchers know how to use it, it has many applications outside of our focus area.

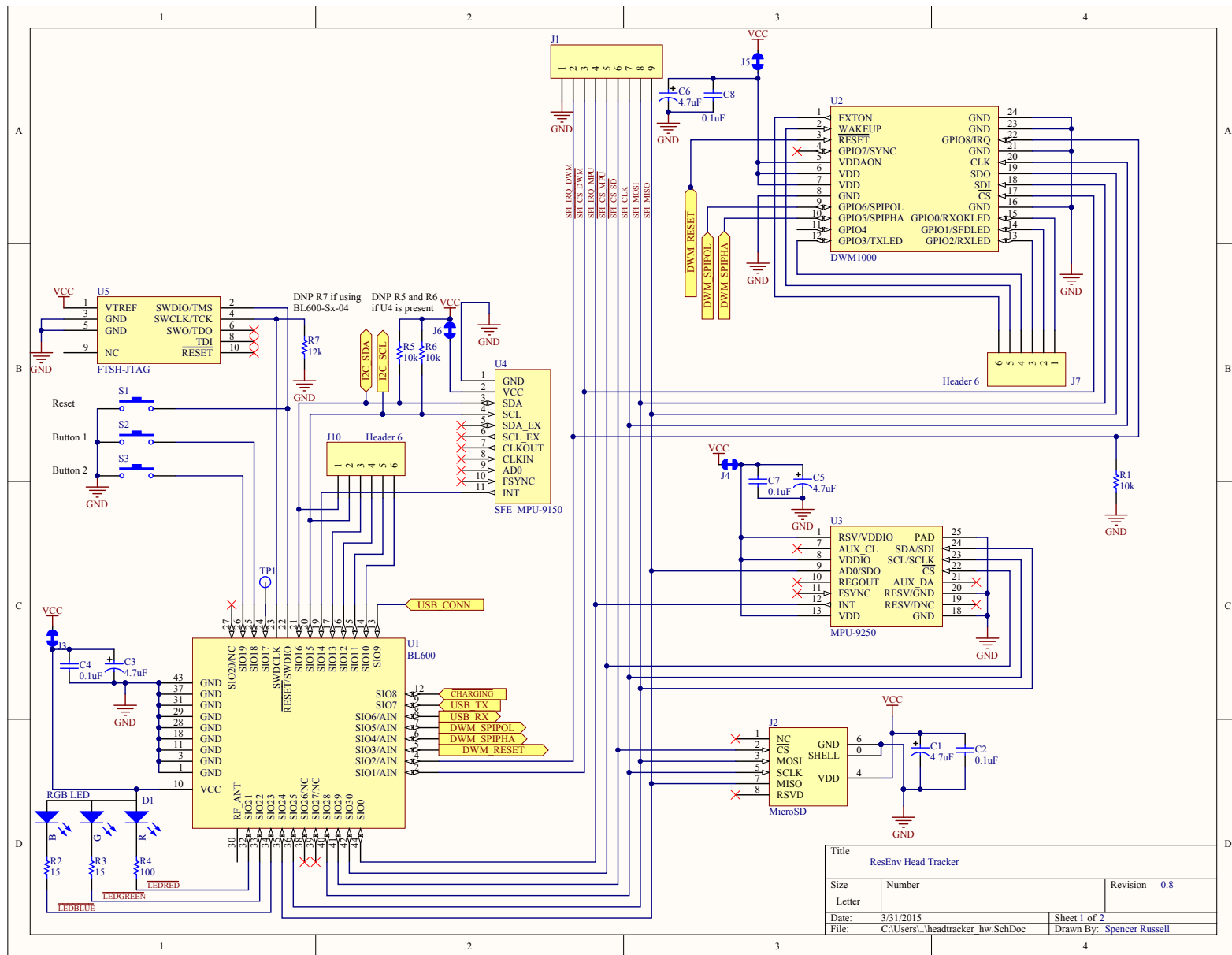
From an engineering perspective there are a number of clear opportunities to improve the performance of HearThere. The development hardware we have built can easily be miniaturized to be less intrusive and more easily mountable. We plan on changing from our current two-way-ranging scheme to a time-difference-of-arrival configuration, which will remove most of the localization load from the tag firmware and allow much more scalability to greater numbers of tags and anchors.

While our particle filter provides good accuracy when it is tracking, clearly there are conditions where it loses the tag. Our probabilistic model is simplistic, and estimating a more sophisticated state space and/or using additional features like step counting may allow us to capture other sources of error, as well as help address non-line-of-site conditions. There are also additional metrics reported by the DecaWave API that should further help identify NLOS conditions.

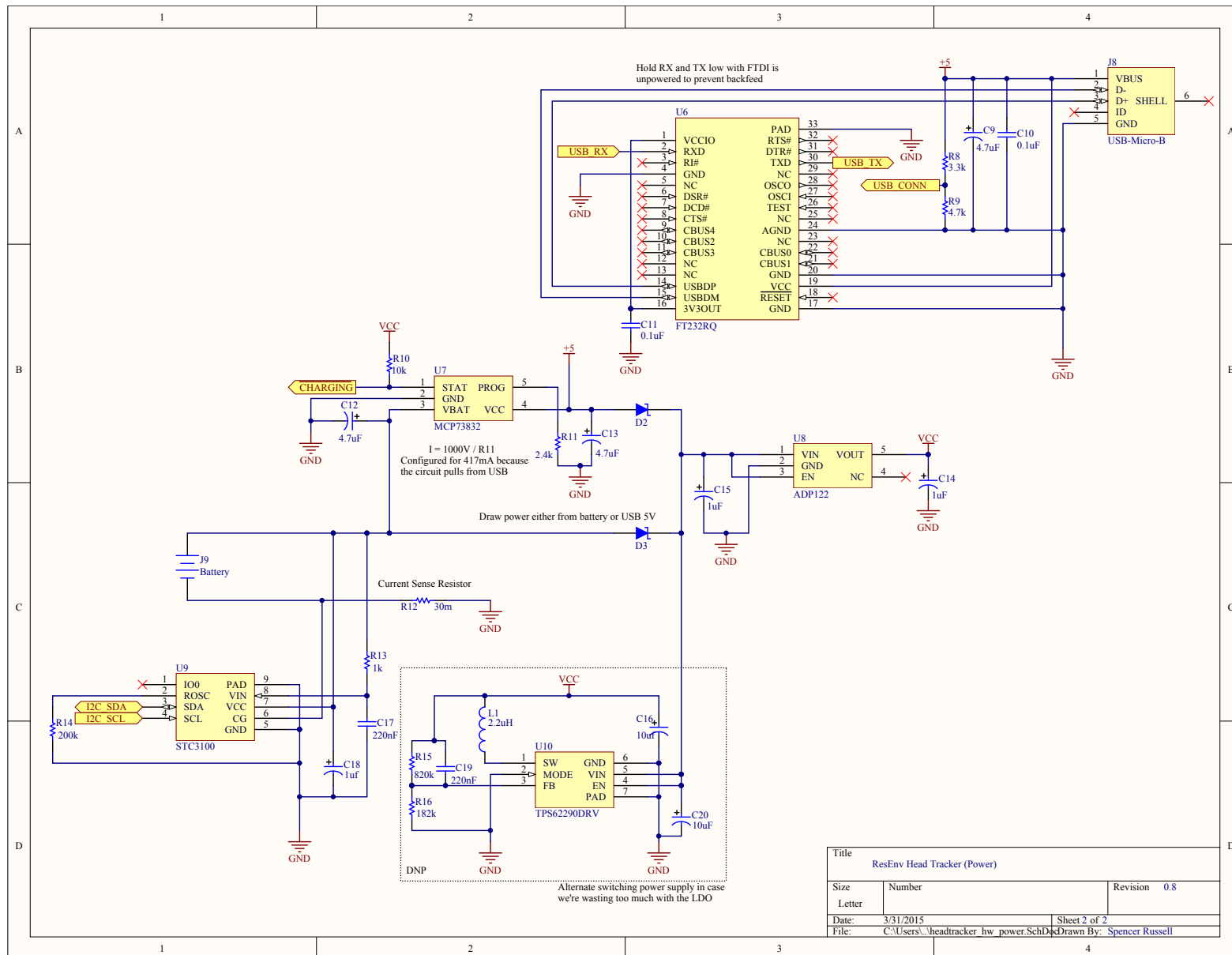
The technology we have built provides the core tools necessary for ubiquitous auditory augmented reality experiences, leaving the most important next steps to be exploring the design space. We are confident that this work forms a strong foundation to build on and look forward to continuing this work.

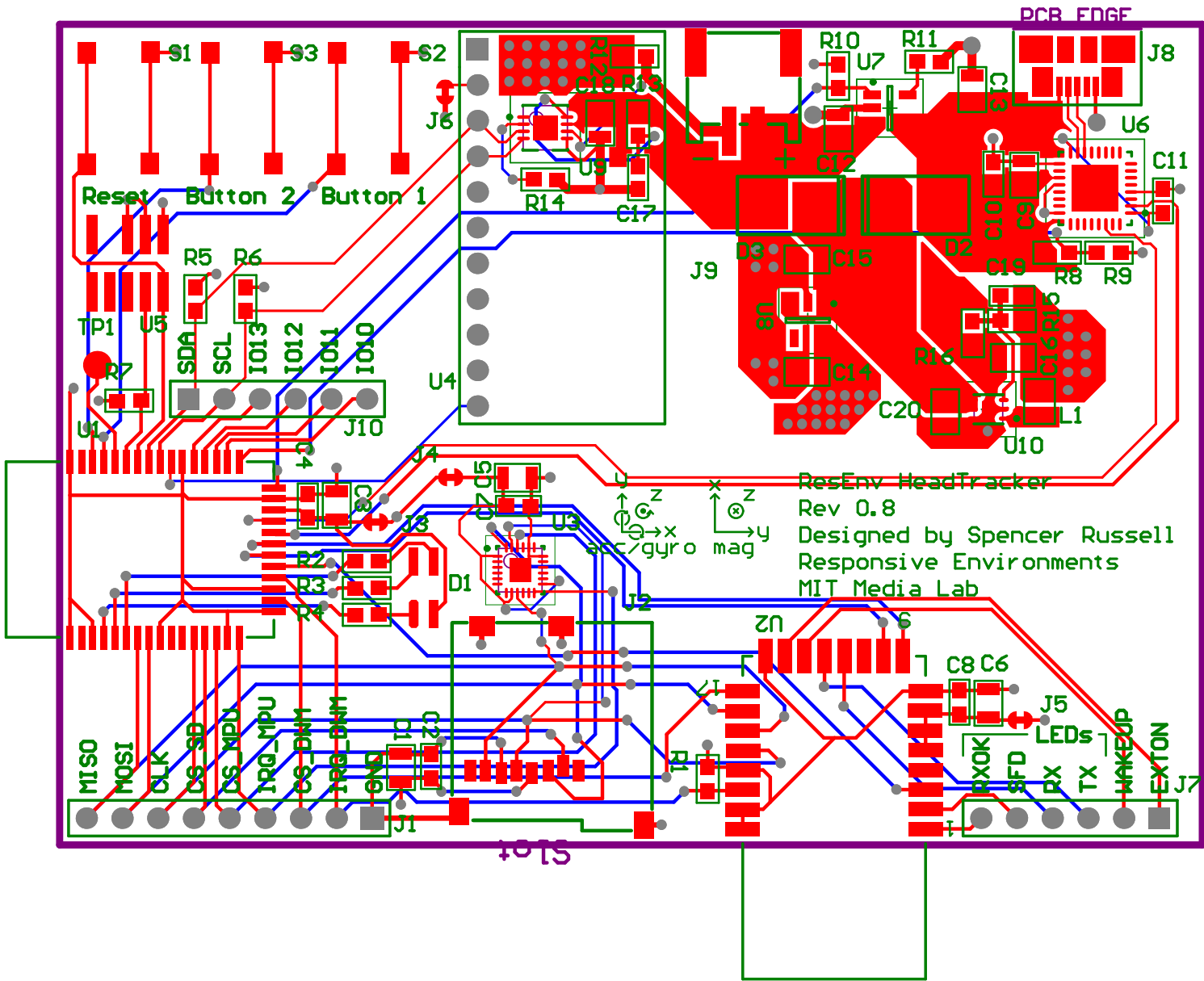


## *Appendix A: Hardware Schematic and Layout*



Title		
ResEnv Head Tracker		
Size	Number	Revision 0.8
Letter		
Date:	3/31/2015	Sheet 1 of 2
File:	C:\Users\...headtracker hw.SchDoc	Drawn By: Spencer Russell





## *Appendix B: Field Test Responses*

After each participant's test we asked them the following survey questions. The full results are listed below.

1. What is your age?

2. How much experience do you have with spatial audio?

The options were

- None
- Novice (listened to surround sound)
- Intermediate (Heard some binaural audio demos)
- Advanced (experimented extensively or authored spatial content)

3. Did you find the task easier with one pair of headphones or the other?

The options were

- Bone Conduction
- No Preference
- Headphones

4. What techniques did you use to locate the sounds?

5. What were the biggest challenges you experienced in trying to locate the sounds?

6. Did you find the sense of integration between the virtual and real sounds to be better with one type of headphones?

The options were

- Bone Conduction
- No Preference
- Headphones

7. At any time were you unsure as to what sounds were real and which were virtual?
8. If you answered Yes to the previous question, which sounds were difficult to distinguish? What techniques did you use to disambiguate them?

### *RockingGoldfish*

#### **Age**

23

#### **How much experience do you have with spatial audio?**

Advanced (experimented extensively or authored spatial content)

#### **Did you find the task easier with one pair of headphones or the other?**

Headphones

#### **What techniques did you use to locate the sounds?**

I scanned angles with my head to try to pick out the direction of a sound and then walked toward it until the level peaked.

#### **What were the biggest challenges you experienced in trying to locate the sounds?**

I found it difficult to determine what direction the sound was coming from especially with the bone conduction. I also found that the epicenter of a source didn't seem entirely fixed.

#### **Did you find the sense of integration between the virtual and real sounds to be better with one type of headphones?**

Bone Conduction

#### **At any time were you unsure as to what sounds were real and which were virtual?**

Yes

#### **If you answered Yes to the previous question, which sounds were difficult to distinguish? What techniques did you use to disambiguate them?**

The bird sounds were somewhat ambiguous although the rate at which their level fell off with distance didn't seem natural. The virtual sounds in general were more localized than I would expect a real sound to be with the bone conduction headphones (maybe the bone conduction headphones do negative compression compared to the headphones). Also, the saxophone recording reverberated in a way inconsistent with the location but this was only noticeable with the isolation headphones.

*FastIbis*

**Age**

30

**How much experience do you have with spatial audio?**

None

**Did you find the task easier with one pair of headphones or the other?**

Bone Conduction

**What techniques did you use to locate the sounds?**

I scanned the area systematically, when it didn't help I walked around randomly, I turned my head around to see where the sound is coming from. I went back and forth to see whether the volume changes.

**What were the biggest challenges you experienced in trying to locate the sounds?**

It was difficult when the sound was not there, or when other sounds were louder. sometimes I would pass a sound and then it would not be there anymore when I come back. for the quite sounds it was hard to tell the direction.

**Did you find the sense of integration between the virtual and real sounds to be better with one type of headphones?**

Bone Conduction

**At any time were you unsure as to what sounds were real and which were virtual?**

Yes

**If you answered Yes to the previous question, which sounds were difficult to distinguish? What techniques did you use to disambiguate them?**

I think it was less the problem that I would confuse the real and virtual sound, rather than expecting the virtual sound to have a real source. When looking for the chickens, I couldn't help but look down as if I was searching a chicken, but for the voice sound I looked straight forward. some sounds caused a reaction as if someone appeared suddenly behind me. I could distinguish virtual from real sound in terms of separating the sound I am looking for.

*SpacyFrog*

**Age**

34

**How much experience do you have with spatial audio?**

None

**Did you find the task easier with one pair of headphones or the other?**

No Preference

**What techniques did you use to locate the sounds?**

Walking in growing circles, walking in grid-like scan. When one of the sounds started to grow stronger I turn to the sides in order to get more accurate direction

**What were the biggest challenges you experienced in trying to locate the sounds?**

some sounds were relatively strong but cut off unexpectedly. Some were “jumpy” (especially the woman voice on the first scenario). Chickens were difficult to find (couldn’t locate them on the second scenario). It’s difficult to explain, but the artificial sounds seemed to come from either right or left, while natural sounds seemed to transition more smoothly as I turned around. Perhaps natural sounds were more consistent/continuous as well

**Did you find the sense of integration between the virtual and real sounds to be better with one type of headphones?**

Bone Conduction

**At any time were you unsure as to what sounds were real and which were virtual?**

No

*StrongCrow*

**Age**

30

**How much experience do you have with spatial audio?**

Novice (listened to surround sound)

**Did you find the task easier with one pair of headphones or the other?**

Headphones

**What techniques did you use to locate the sounds?**

I rotated my head when I heard something distinctively coming from the left or right, to orient the sound so it felt like the sound direction was in front of me. i then took a few steps forward, slowly. if the sound started skewing to the left, or right, i corrected



and repeated until the sound was very loud.

**What were the biggest challenges you experienced in trying to locate the sounds?**

ambient noise was too loud. the woman talking was too soft. there was some delay, so i noticed when i moved too quickly a sound would get really loud and then disappear. then i would have to back track to find the source of the sound. sometimes sounds would come and go seemingly randomly, which confused me about where the sound was coming from.

**Did you find the sense of integration between the virtual and real sounds to be better with one type of headphones?**

Headphones

**At any time were you unsure as to what sounds were real and which were virtual?**

Yes

**If you answered Yes to the previous question, which sounds were difficult to distinguish? What techniques did you use to disambiguate them?**

the birds were difficult to disambiguate, but they were high pitched and loud, which made it a bit easier to find (as opposed to the female voice, for example).

*LostCobra*

**Age**

36

**How much experience do you have with spatial audio?**

Intermediate (Heard some binaural audio demos)

**Did you find the task easier with one pair of headphones or the other?**

No Preference

**What techniques did you use to locate the sounds?**

Walk around the area in circle one way or two. Try to find the target one by one from the easiest one. Remember the approximate sound level of each targets and move to the location where I heard the sound largest. Compare L/R and move to either way.

**What were the biggest challenges you experienced in trying to locate the sounds?**

Interference. Absolute sound level of the sources seemed different and the larger one often masked the smaller ones. I simply forgot the level or direction of the sound source while listening to several

sounds. Sudden L/R change of the source. I was confused about if I got close to the sound or not.

**Did you find the sense of integration between the virtual and real sounds to be better with one type of headphones?**

No Preference

**At any time were you unsure as to what sounds were real and which were virtual?**

Yes

**If you answered Yes to the previous question, which sounds were difficult to distinguish? What techniques did you use to disambiguate them?**

Birds (only at a short time) I knew that the virtual sounds came continuously so I waited for the next que of the sounds

*IndigoCheetah*

**Age**

30

**How much experience do you have with spatial audio?**

Intermediate (Heard some binaural audio demos)

**Did you find the task easier with one pair of headphones or the other?**

Headphones

**What techniques did you use to locate the sounds?**

Closing my eyes helped me “visualize” where the sounds might be coming from. I was concerned that my view of the transmitters were altering my perception of volume with signal strength. Trying to focus on a single sound at a time and then navigate towards it helped me to isolate sounds.

**What were the biggest challenges you experienced in trying to locate the sounds?**

I couldn't hear the chickens through the bone conduction headphones. I also spent some time navigating a bit like a Roomba because the sounds dropped off from their locations faster than I imagine in real space. When I was close to the woman or saxophone, I could hear them quite loudly, but I imagine I would still be able to hear them in the space if I wasn't next to them. Additionally, the high pitch of the birds noises were difficult to get direction on. Lastly, having the sounds be intermittent (i.e. sometimes the sax would stop) threw me off a bit.

**Did you find the sense of integration between the virtual and**

**real sounds to be better with one type of headphones?**

Headphones

**At any time were you unsure as to what sounds were real and which were virtual?**

Yes

**If you answered Yes to the previous question, which sounds were difficult to distinguish? What techniques did you use to disambiguate them?**

At one point a real bird flew over head and sang and my head turned to see it nearly instantly. While for a moment I thought it might have been virtual, I was quickly confirmed otherwise.

## References

- Azuma, Ronald T. et al. "A survey of augmented reality". In: *Presence* 6.4 (1997), pp. 355–385. URL: [http://www.mitpressjournals.org/userimages/ContentEditor/1332945956500/PRES\\_6-4\\_Azuma\\_web.pdf](http://www.mitpressjournals.org/userimages/ContentEditor/1332945956500/PRES_6-4_Azuma_web.pdf) (visited on 08/16/2015).
- Begault, Durand R., Elizabeth M. Wenzel, and Mark R. Anderson. "Direct comparison of the impact of head tracking, reverberation, and individualized head-related transfer functions on the spatial perception of a virtual speech source". In: *Journal of the Audio Engineering Society* 49.10 (2001), pp. 904–916. URL: <http://www.aes.org/e-lib/browse.cfm?elib=10175> (visited on 07/31/2015).
- Bezanson, Jeff et al. "Julia: A Fresh Approach to Numerical Computing". In: (Nov. 2014).
- Blum, Jeffrey R., Mathieu Bouchard, and Jeremy R. Cooperstock. "What's around me? Spatialized audio augmented reality for blind users with a smartphone". In: *Mobile and Ubiquitous Systems: Computing, Networking, and Services*. Springer, 2012, pp. 49–62. URL: [http://link.springer.com/chapter/10.1007/978-3-642-30973-1\\_5](http://link.springer.com/chapter/10.1007/978-3-642-30973-1_5) (visited on 07/31/2015).
- Brimijoin, W Owen and Michael A Akeroyd. "The role of head movements and signal spectrum in an auditory front/back illusion". en. In: *i-Perception* 3.3 (2012), pp. 179–181. ISSN: 2041-6695. DOI: 10.1068/i7173sas. URL: <http://i-perception.perceptionweb.com/journal/I/article/i7173sas> (visited on 07/31/2015).
- Chapin, W. L. "InTheMix". In: Siggraph, 2000. URL: <http://www.ausim3d.com/InTheMix/>.
- Chiu, David S. and Kyle P. O'Keefe. "Seamless outdoor-to-indoor pedestrian navigation using GPS and UWB". In: *Proceedings of the 21st International Technical Meeting of the Satellite Division of the Institute of Navigation (ION GNSS 2008), The Institute of Navigation*. Vol. 1. 2008, pp. 322–333. URL: [http://plan.geomatics.ucalgary.ca/papers/chiuokeefe\\_uwbindoor\\_gnss2008\\_web.pdf](http://plan.geomatics.ucalgary.ca/papers/chiuokeefe_uwbindoor_gnss2008_web.pdf) (visited on 07/31/2015).

- Chung, Jaewoo et al. "Indoor location sensing using geo-magnetism". In: *Proceedings of the 9th international conference on Mobile systems, applications, and services*. ACM, 2011, pp. 141–154. URL: <http://dl.acm.org/citation.cfm?id=2000010> (visited on 08/15/2015).
- Gezici, S. et al. "Localization via ultra-wideband radios: a look at positioning aspects for future sensor networks". In: *IEEE Signal Processing Magazine* 22.4 (July 2005), pp. 70–84. ISSN: 1053-5888. DOI: 10.1109/MSP.2005.1458289.
- Gonzalez, Jose et al. "Combination of UWB and GPS for indoor-outdoor vehicle localization". In: *Intelligent Signal Processing, 2007. WISP 2007. IEEE International Symposium on*. IEEE, 2007, pp. 1–6. URL: [http://ieeexplore.ieee.org/xpls/abs\\_all.jsp?arnumber=4447550](http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4447550) (visited on 08/16/2015).
- González, J. et al. "Mobile robot localization based on Ultra-Wide-Band ranging: A particle filter approach". en. In: *Robotics and Autonomous Systems* 57.5 (May 2009), pp. 496–507. ISSN: 09218890. DOI: 10.1016/j.robot.2008.10.022. URL: <http://linkinghub.elsevier.com/retrieve/pii/S0921889008001747> (visited on 07/31/2015).
- Härmä, Aki et al. "Augmented reality audio for mobile and wearable appliances". In: *Journal of the Audio Engineering Society* 52.6 (2004), pp. 618–639. URL: <http://www.aes.org/e-lib/browse.cfm?elib=13010> (visited on 07/31/2015).
- Hightower, Jeffrey and Gaetano Borriello. "Location systems for ubiquitous computing". In: *Computer* 8 (2001), pp. 57–66. URL: <http://www.computer.org/csdl/mags/co/2001/08/r8057.pdf> (visited on 09/01/2015).
- Huynh, Du Q. "Metrics for 3D rotations: Comparison and analysis". In: *Journal of Mathematical Imaging and Vision* 35.2 (2009), pp. 155–164. URL: <http://link.springer.com/article/10.1007/s10851-009-0161-2> (visited on 08/12/2015).
- Klatzky, Roberta L. et al. "Cognitive load of navigating without vision when guided by virtual sound versus spatial language." en. In: *Journal of Experimental Psychology: Applied* 12.4 (2006), pp. 223–232. ISSN: 1939-2192, 1076-898X. DOI: 10.1037/1076-898X.12.4.223. URL: <http://doi.apa.org/getdoi.cfm?doi=10.1037/1076-898X.12.4.223> (visited on 07/31/2015).
- Kuipers, Jack B. *Quaternions and rotation sequences*. Vol. 66. Princeton university press Princeton, 1999. URL: <http://www.emis.ams.org/proceedings/Varna/vol1/GEOM09.pdf> (visited on 08/04/2015).
- Li, Wu-Hsi. "Loco-Radio: designing high-density augmented reality audio browsers". PhD thesis. Massachusetts Institute of

- Technology, 2013. URL: <http://dspace.mit.edu/handle/1721.1/91876> (visited on 08/15/2015).
- MacDonald, Justin A., Paula P. Henry, and Tomasz R. Letowski. "Spatial audio through a bone conduction interface: Audición espacial a través de una interfase de conducción ósea". en. In: *International Journal of Audiology* 45.10 (Jan. 2006), pp. 595–599. ISSN: 1499-2027, 1708-8186. DOI: 10.1080/14992020600876519. URL: <http://informahealthcare.com/doi/abs/10.1080/14992020600876519> (visited on 07/31/2015).
- MacKenzie, Donald A. *Inventing accuracy: A historical sociology of nuclear missile guidance*. MIT press, 1993. URL: <https://books.google.com/books?hl=en&lr=&id=QymEXZIWEe8C&oi=fnd&pg=PR9&dq=%22inventing+accuracy%22&ots=qcS4DLIXYZ&sig=4QtKXkhYrMttFu6Fl099VqTvY5M> (visited on 09/01/2015).
- Madgwick, Sebastian OH. "An efficient orientation filter for inertial and inertial/magnetic sensor arrays". In: *Report x-io and University of Bristol (UK)* (2010). URL: [http://sharenet-wii-motion-trac.googlecode.com/files/An\\_efficient\\_orientation\\_filter\\_for\\_inertial\\_and\\_inertialmagnetic\\_sensor\\_arrays.pdf](http://sharenet-wii-motion-trac.googlecode.com/files/An_efficient_orientation_filter_for_inertial_and_inertialmagnetic_sensor_arrays.pdf) (visited on 07/31/2015).
- Madgwick, Sebastian OH, Andrew JL Harrison, and Ravi Vaidyanathan. "Estimation of IMU and MARG orientation using a gradient descent algorithm". In: *Rehabilitation Robotics (ICORR), 2011 IEEE International Conference on*. IEEE, 2011, pp. 1–7. URL: [http://ieeexplore.ieee.org/xpls/abs\\_all.jsp?arnumber=5975346](http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5975346) (visited on 07/31/2015).
- Mautz, Rainer. "Indoor positioning technologies". Habilitation Thesis. ETH Zürich, 2012, 2012. URL: <http://e-collection.library.ethz.ch/view/eth:5659> (visited on 08/15/2015).
- Mayton, Brian. "WristQue: A Personal Sensor Wristband for Smart Infrastructure and Control". MA thesis. Massachusetts Institute of Technology, 2012.
- McElroy, Carter, Dries Neiryneck, and Michael McLaughlin. "Comparison of wireless clock synchronization algorithms for indoor location systems". In: *Communications Workshops (ICC), 2014 IEEE International Conference on*. IEEE, 2014, pp. 157–162. URL: [http://ieeexplore.ieee.org/xpls/abs\\_all.jsp?arnumber=6881189](http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6881189) (visited on 07/31/2015).
- McEwan, T. and S. Azevedo. "Micropower impulse radar". In: *Science & Technology Review* (1996).
- McEwan, Thomas E. "Ultra-wideband receiver". US5345471 A. Sept. 1994. URL: <http://www.google.com/patents/US5345471> (visited on 09/01/2015).

- Mika, I. et al. "Optical positioning and tracking system for a head mounted display based on spread spectrum technology". In: 1998, pp. 597–608. ISBN: 952-15-0036-0. URL: <http://cat.inist.fr/?aModele=afficheN%5C&cpsid=1368510> (visited on 08/02/2015).
- Minnaar, Pauli et al. "The importance of head movements for binaural room synthesis". In: (2001). URL: <https://smartech.gatech.edu/handle/1853/50656> (visited on 08/01/2015).
- Mynatt, Elizabeth D. et al. "Designing audio aura". In: *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM Press/Addison-Wesley Publishing Co., 1998, pp. 566–573. URL: <http://dl.acm.org/citation.cfm?id=274720> (visited on 08/15/2015).
- OlliW. *IMU Data Fusing: Complementary, Kalman, and Mahony Filter*. [Online; accessed 5-November-2014]. Sept. 2013. URL: <http://www.olliw.eu/2013/imu-data-fusing/>.
- Palovuori, Karri T., Jukka J. Vanhala, and Markku A. Kivikoski. "Shadowtrack: A Novel Tracking System Based on Spread-Spectrum Spatio-Temporal Illumination". In: *Presence: Teleoperators & Virtual Environments* 9.6 (Dec. 2000), pp. 581–592. ISSN: 10547460. DOI: 10.1162/105474600300040394. URL: <http://libproxy.mit.edu/login?url=http://search.ebscohost.com/login.aspx?direct=true%5C&db=a9h%5C&AN=4278901%5C&site=ehost-live> (visited on 08/02/2015).
- Raskar, Ramesh et al. "Prakash: lighting aware motion capture using photosensing markers and multiplexed illuminators". In: *ACM Transactions on Graphics (TOG)*. Vol. 26. ACM, 2007, p. 36. URL: <http://dl.acm.org/citation.cfm?id=1276422> (visited on 08/01/2015).
- Ross, Gerald F. "Transmission and reception system for generating and receiving base-band pulse duration pulse signals without distortion for short base-band communication system". US3728632 A. Apr. 1973. URL: <http://www.google.com/patents/US3728632> (visited on 09/01/2015).
- Sandvad, J. "Dynamic Aspects of Auditory Virtual Environments". In: *Audio Engineering Society Convention 100*. May 1996. URL: <http://www.aes.org/e-lib/browse.cfm?elib=7547>.
- Schmitt, R. et al. "Performance evaluation of iGPS for industrial applications". In: *2010 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*. Sept. 2010, pp. 1–8. DOI: 10.1109/IPIN.2010.5647630.
- Sources of Error in DW1000 Based Two-Way Ranging (TWR) Schemes*. Application Note APS011. DecaWave, 2014.

- Thrun, Sebastian, Wolfram Burgard, and Dieter Fox. *Probabilistic Robotics*. English. Cambridge, Mass: The MIT Press, Aug. 2005. ISBN: 978-0-262-20162-9.
- Thurlow, Willard R. and Philip S. Runge. "Effect of induced head movements on localization of direction of sounds". In: *The Journal of the Acoustical Society of America* 42.2 (1967), pp. 480–488. URL: <http://scitation.aip.org/content/asa/journal/jasa/42/2/10.1121/1.1910604> (visited on 08/16/2015).
- Townsend, Kevin et al. *Getting Started with Bluetooth Low Energy: Tools and Techniques for Low-Power Networking*. English. 1 edition. O'Reilly Media, Apr. 2014.
- Walker, Bruce N. and Jeffrey Lindsay. "Navigation performance in a virtual environment with bonephones". In: (2005). URL: <https://smartech.gatech.edu/handle/1853/50173> (visited on 07/31/2015).
- Wallach, Hans. "The role of head movements and vestibular and visual cues in sound localization." In: *Journal of Experimental Psychology* 27.4 (1940), p. 339. URL: <http://psycnet.apa.org/journals/xge/27/4/339/> (visited on 08/16/2015).
- Wilson, Jeff et al. "Swan: System for wearable audio navigation". In: *Wearable Computers, 2007 11th IEEE International Symposium on*. IEEE, 2007, pp. 91–98. URL: [http://ieeexplore.ieee.org/xpls/abs\\_all.jsp?arnumber=4373786](http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4373786) (visited on 08/16/2015).
- Win, M.Z. et al. "History and Applications of UWB [Scanning the Issue]". In: *Proceedings of the IEEE* 97.2 (Feb. 2009), pp. 198–204. ISSN: 0018-9219. DOI: 10.1109/JPROC.2008.2008762.
- Yiu, Joseph. *The Definitive Guide to the ARM Cortex-M0*. en. Elsevier, Apr. 2011. ISBN: 978-0-12-385478-0.
- Zimmermann, Andreas, Andreas Lorenz, and S. Birlinghoven. "LISTEN: Contextualized presentation for audio-augmented environments". In: *Proceedings of the 11th Workshop on Adaptivity and User modeling in Interactive Systems*. 2003, pp. 351–357. URL: <https://km.aifb.kit.edu/ws/LLWA/abis/Zimmermann.pdf> (visited on 08/01/2015).