Department of Electronics

MEng Project Report

2015/2016

Student Name: Lewis Thresh

Project Title: Using the Direct Room Impulse Response Rendering Method to Enable Mobility Within A Virtual Acoustic Environment and Assessing Its Plausibility

Supervisors: Dr. Jude Breteron & Mr Marc Green

Department of Electronics
University of York
Heslington
York
YO10 5DD

Acknowledgements

I would like to first of all thank my supervisor Dr Jude Brereton for her guidance, support and continuous reassurance throughout the duration of this project.

I would also like to thank Andrew Chadwick for his contribution through useful conversations and knowledgeable insight.

Finally, I would like to thank my parents, Paul and Joanna Thresh, for encouraging me in whatever I do and supporting me regardless of the outcome.

Statement of Ethics

As test participants were required during this project, the ethics of doing so were considered and the appropriate ethics forms were submitted.

Supporting Material

A supporting web page for the work described in this document can be found here. All code, audio sample and videos mentioned in the text can be viewed/heard by clicking on the file name provided.

All relevant data produced as a result of this project, including that found on the supporting webpage can be located within the appropriate file within the following file structure:

```
Dissertation.pdf
└─ 0) SupportingFiles
    ├─ 1) Audio
    │   ├─ 1.1) Anechoic Recordings
    │   ├─ 1.2) Calibrated RIRs
    │   ├─ 1.3) Odeon Settings
    │   └─ 1.4) Odeon Materials
    ├─ 2) Matlab
    ├─ 3) MaxPatch
    │   ├─ 3.1) Patch
    │   └─ 3.2) Data
    └─ 4) Room Model
        ├─ 4.1) SketchUp
        ├─ 4.2) Odeon
        │   ├─ 4.2.1) Odeon project
        │   └─ 4.2.2) Source scripts
        └─ 4.3) Material List
```

For example, if the text were to read:

...the relavant code can be found in file 3.1...

the file in question can be found in the `Patch` folder.

ABSTRACT

The project described in this document builds upon the Virtual Singing Studio, a loudspeaker based room acoustic simulation system, by extending its functionality to allow a user to move themselves around a virtual space through the use of a large grid of synthetically produced room impulse responses. The process of implementing said system was assessed and its plausibility investigated. Though the system in place provides a framework to build upon, it was found that for a real time application, the system in question encounters too many technical difficulties to be considered accurate or a beneficial tool.

CONTENTS

1 Introduction **1**

2 Background **3**

2.1 Virtual Acoustic Environments . . . 3
2.2 Room Impulse Responses . . . 3
2.3 Ambisonics . . . 4
2.4 Software Overview . . . 5
2.4.1 Odeon: Simulating Room Acoustics . . . 5
Ray-Tracing . . . 5
Image Source Method . . . 6
Hybrid Method . . . 7
Scattering . . . 8
Room Modelling . . . 8
Material Selection . . . 8
2.4.2 Google SketchUp . . . 9
2.4.3 Max/MSP . . . 9
2.5 Project Description . . . 9
2.5.1 The Virtual Singing Studio . . . 9
2.5.2 Project Aims and Motivation . . . 10
2.5.3 Project Objectives . . . 13

3 Implementation **14**

3.1 Room Choice . . . 14
3.2 User Test Planning . . . 14
3.3 Real RIR Recordings . . . 15
3.3.1 RIR Measurement Set Up . . . 15
3.3.2 Positions . . . 15
3.3.3 Measurements . . . 17
3.3.4 Measurements Post-Production . . . 17
3.3.5 Issues . . . 18
3.4 Room Modelling . . . 18
3.4.1 Room Measurements . . . 18
3.4.2 Designing the room . . . 18
3.5 ODEON . . . 20
3.5.1 Water Tightness Test . . . 20
3.5.2 Material Selection . . . 21
Initial Materials . . . 21

Surface Types 22
RIR Comparison 24
3.5.3 Odeon Output Settings 28
Ray Settings 28
Directivity 32
3.5.4 RIR Topology Problems 33
RIR Analysis 33
Experiment 35
Results 36
3.5.5 RIR Locations 39
3.5.6 RIR Rendering 40
3.6 Producing Different RIR Grids 41
3.7 RIR Calibration 42
3.7.1 Matlab 42
Method 42
Issue 46
3.7.2 Reaper 47
3.8 Room Modelling and Synthetic RIR Rendering Summary 47
3.9 Software 48
3.9.1 Software Overview 48
The Original Patch 48
Extended Patch Overview 48
3.9.2 Location Selection 50
File name JavaScript: 'loadFilesLogic.js' 54
3.9.3 Mobility Implementation 56
Iteration 1 56
Iteration 2 57
Iteration 3 (Final) 58
3.9.4 Location Tracking 62
User Interface 62
Position Feedback 63
3.9.5 Head-Tracking 64
3.9.6 Settings Menu 65
3.9.7 Software Issues 65
Location approximation 65
Travel Speed 66
3.10 Latency 67
3.11 RIR Trimming 69
3.12 Software Implementation Summary 69

4 User Testing **71**
4.1 User Testing Aims 71

4.2 Test #1 72
4.2.1 Procedure 72
4.2.2 Results 72
4.2.3 Discussion 73
4.3 Test #2 74
4.3.1 Procedure 74
4.3.2 Results 75
4.3.3 Discussion 77
4.4 Test #3 77
4.4.1 Procedure 77
4.4.2 Results 78
4.4.3 Discussion 79
4.5 User Tests Discussion and Summary 80
4.5.1 Test Length 80
4.5.2 The 'I Don't Know' Problem 80
4.5.3 Results Summary 81

5 Conclusion **82**
5.1 Project Summary 82
5.2 Project Success Evaluation 82
5.3 Overall System Implementation 83
5.4 Final Comments 84

6 Further Work **85**
6.1 Software 85
6.1.1 Storage 85
6.1.2 Iteration 1 Redesign 85
6.1.3 UI cross platform 85
6.2 Accuracy 86
6.2.1 RIR Modelling 86
6.2.2 Surface Materials 86
6.3 Additional Functionality 86

7 Project Management **88**
7.1 Management Methods 88
7.2 Dealing With Issues 88

Appendices **92**

A Appendix A **93**

B Appendix B **97**

C Appendix C **99**

D Appendix D **100**

E Appendix E **106**

Acronyms

VAE Virtual Acoustic Environment 3

VSS Virtual Singing Studio 1

RIR Room Impulse Response 3

ISM Image Source Method 5

TO Transition Order 7

VBS Vector Based Scattering Method 28

LR's Late Rays 29

Introduction

No prior understanding of room acoustics is required to appreciate the difference in sound experienced when talking, singing or playing an instrument in rooms of different size and interior. For those who listen even more closely, the effect of standing in different positions of a room can also be appreciated, such as the deep booming sound of standing in a corner or the fluttering echo heard when clapping in the centre of two symmetrical walls. However, the benefits of the hard work that has gone into studying room acoustics and sound propagation are apparent across multiple platforms, from understanding how to acoustically treat a room to make it sound the way we want, to creating tools and defining methods that allow us to recreate or synthesis a rooms acoustics, leading to the ability to develop realistic soundscapes that can bring virtual worlds to life, such as in video games and films.

The ability to do such things has been utilised in other areas of research, such as studying how musicians perform differently when playing in rooms with different room acoustics [1]. This has been done at the University of York, where the benefit of being able to simulate the acoustics of multiple rooms within moments has been used to effectively transport musicians and researchers to different venues without having to travel anywhere. The system used to do this is the Virtual Singing Studio (VSS), upon which this project is based. In addition to allowing a user to simply hear themselves in different venues, this project aims to allow the user to also move themselves around that venue, allowing them to experiment with performing in different positions.

The rest of this document will be structured as follows:

Background
An overview of all the material that form the basis of this project, including an overview of the software used to achieve the end result. Previous work is discussed, leading to a full description of the project including the motivation behind it. Finally, the project aims and objectives are stated, including how the system was to be produced and tested to access plausibility, providing a set a statements that are used to measure the success of the project upon completion.

Implementation
This section presents the practical work that was carried out throughout the project, describing the decisions made, issues encountered and how they were overcome.

User Testing
The procedure that was carried out to test the implemented system is described and the results are presented and discussed, followed by a conclusion, highlighting potential reasons for the obtained results.

Project Management
A brief overview of how the project was planned and time-managed.

Conclusion

The project as a whole is summarised and measured against the initial metrics set, evaluating the overall success of the project. The issues raised throughout previous sections are reviewed and potential solutions are discussed. Further improvements to the system are suggested, including some of the work that was initially intended to be carried out.

Background

The following section covers material that forms the basis of the system produced as a result of this project.

2.1 - Virtual Acoustic Environments

Virtual acoustics has been previously described [2] as follows:

> *"Virtual acoustics is a general term for the modelling of acoustical phenomena and systems with the aid of a computer"*

By this definition, a Virtual Acoustic Environment (VAE) can be thought of as an environment (such as a room) for which the acoustical phenomena have been either recreated or synthesised. To produce a VAE, prior knowledge regarding the room which is to be acoustically recreated must be known; how do all audible frequencies propagate around the room for a set sound source location and receiver location?

This information can be gathered by taking a Room Impulse Response (RIR) and used to recreate the acoustics of a room for the set sound source and receiver location.

2.2 - Room Impulse Responses

In order to reproduce the acoustical phenomena of a room, an RIR must be obtained. This is done by exciting all audible frequencies within the room by using a sound source such as a loudspeaker, and recording the result using a receiver microphone.

There are a number of techniques used for exciting all audible frequencies. These include an **impulse** (such as a starter pistol) or an **exponentially swept sine** for which a sine wave is exponentially increased in frequency over a fixed period of time. The difference between these techniques is that an impulse emits all of the audible frequencies into the room at the same time, whereas the sine sweep does it gradually. As the idea is to obtain an **impulse**, using the impulse technique is far simpler as no post processing is required. Using an exponentially swept sine requires post processing in order to time align all of the frequency dependent room reflections through the use of a deconvolution algorithm, making this method a little less simple. However, this method produces a greater signal to noise ratio thus is the desired method [3]. Once the impulse is obtained, it can be convolved with an audio signal, making that audio signal sound as though it is being produced within the room that the RIR was taken from. Convolution simply takes an input signal and multiplies each of the discrete samples with the impulse response, thus simulating the effect the room would have on each of these samples if they were actually emitted in the room itself [4].

2.3 - Ambisonics

Though using an omni-directional microphone to record an RIR is a set standard [5], it is also possible to use techniques such as Ambisonics to record RIR's that can be used to reproduce a soundfield in three dimensions.

Ambisonics is a technique used to encode and decode three dimensional spatial audio information using just four audio channels. A three dimensional sound field can be recorded using a microphone known as a Soundfield microphone shown on the left in figure 1. These microphones contain four coincident capsules, one of which is an omni-directional capsule (W) and the rest of which are figure of 8 capsules used to record sound in the X (front and back), Y (left and right) and Z (up and down) direction illustrated on the right in figure 1. In theory, the aim is to record the sound field at a single point, however this will never be possible given that the microphones take up physical space. Therefore, the sound captured by the microphones are first recorded into A-format (raw unedited signals) and then converted to B-format which is the same signal with an applied inter-capsule time correction to compensate for the physical separation of the microphones [6] and the fact that they are not absolutely coincident [7]. Once the sound field has been captured, a system specific decoder can be used to replay the captured signal over an arbitrary number of loudspeakers.

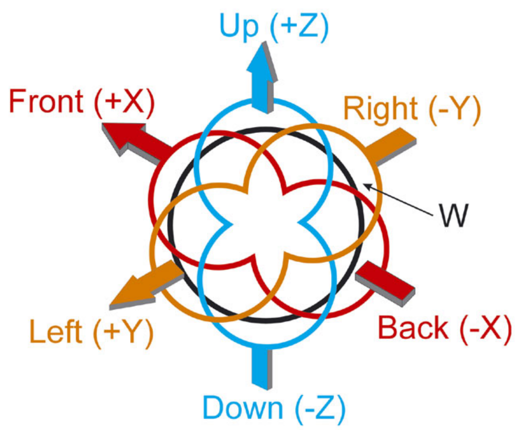


Figure 1: **Left**: Picture of a Soundfield microphone with coincident capsules exposed. **Right**: Soundfield microphone polar pattern. Images sourced from [8] (background removed from left image)

2.4 - Software Overview

2.4.1 Odeon: Simulating Room Acoustics

Methods for physically measuring RIR's within rooms has been discussed, however there is also a way to synthesis RIR's by mathematical modelling the way in which sound interacts with a room. The two methods of doing so are described as **Wave-based Methods** and **Geometrical Acoustic Methods**.

Note: The next paragraph on wave-based methods is adapted from the initially submitted literature review and is not required for the understanding of this project. However, it is provided here as the initial project plan included the use of such methods and is therefore mentioned later in this section as well as in section Further Work

Wave-based methods are based on solving the wave equation, producing a solution that is physically correct and can be used to accurately model how a sound would propagate from a source in a room and reflect [9]. One such example is the finite-difference time-domain (FDTD) method, where sound is modelled by calculating the interaction between nodes in a rectangular mesh. The number of nodes in the mesh is determined by the frequency being modelled, therefore as the frequency being modelled increases, the number of nodes required in the mesh also increases which in turn increases computation time. It is for this reason that using wave-based methods are impractical for calculating high frequency sound propagation.

Instead, it is possible to synthesis RIR's by using room acoustic simulation software such as Odeon [10] that utilises the much quicker geometrical methods. Odeon was designed to provide reliable predictions of room acoustics by using a hybrid of two geometric acoustic modelling methods: **ray-tracing** and the **Image Source Method (ISM)** to synthesis RIR's. These methods model how sound would propagate and interact with a room as though it were a straight line (ray). This inherently neglects wave phenomena such as phase and diffraction, properties that are negligible at high frequencies, however they are fundamental in describing low frequency wave behaviour [9]. Therefore geometrical methods are not accurate at modelling sound propagation for low frequency waves, however they provide a good approximation for most applications and are much quicker to compute.

A description of both of the geometrical methods are described below, leading to a description of the hybrid method used in Odeon.

Ray-Tracing

The ray-tracing method imitates a sound source by emitting a large number of particles in various directions from a single point [11]. These particles are then traced around the room, losing energy each time the particle encounters a surface by a factor of $1 - \alpha$, where α is the absorption coefficient assigned to that surface, illustrated in figure 2. The angle at which the particle is then reflected is determined by the scattering coefficient assigned to the surface of contact, ranging from a specular reflection to a completely random reflection [12] (described in section Scattering). For a specific receiver position, an area around said point is defined in which rays are collected and used to calculate the results.

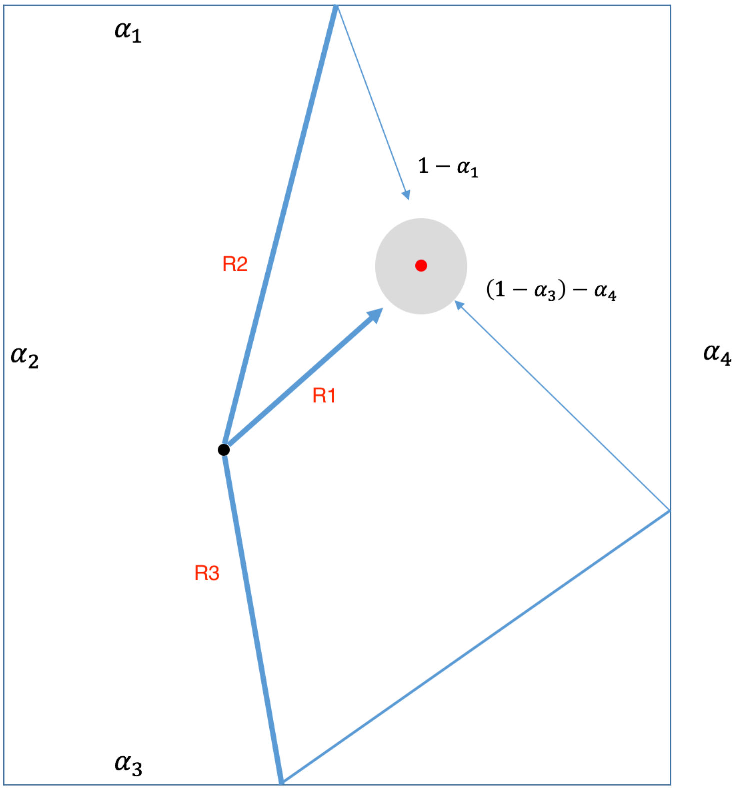


Figure 2: Illustration of the ray-tracing method (by the author) showing a sound source (black) emitting three rays (R1, R2 and R3) and shows how each ray is attenuated by a factor of $(1-\alpha$ each time it is reflected from a wall. The red dot represents the receiver positions and the outlining gray circle represents the area in which rays need to pass through to be used in the RIR calculation.

Ray-tracing does not provide a completely accurate result as it is a risk that some rays may not pass close enough to the receiver to contribute to the final result. The outcome of the ray-tracing method is a statistical result rather than a complete one.

Image Source Method

The ISM can be used to find all possible specular reflection paths from the sound source to the receiver position. This is done by representing each specular reflection (a reflection with angle equal to the angle of incidence) from a surface as a secondary source known as an "image source" [11], illustrated in figure 3. Once these image sources have been created, a visibility test is run which checks to see which image sources are in sight of the receiver thus calculating which image sources should be used in the final calculation of the RIR. If the receiver is then moved, the images sources do not need to be recalculated, only the visibility check needs to be run again. It is due to this that the ISM is much more accurate than ray-tracing which relies on random reflections to calculate an RIR. However, as each new image source emits a number of rays of its own, there is a potential for it to produce a number of image sources. Therefore, with each new image source there is an exponential growth in the total number of image sources, making this method much more computationally expensive than ray-tracing. This problem can be avoided by setting a *reflection order* which determines how many times a ray can reflect around a room before calculations are stopped, thus preventing the creation of more image sources. This however will cause the prediction of the rooms acoustics to be incomplete. This obviously then makes the ISM less accurate with a smaller reflection order.

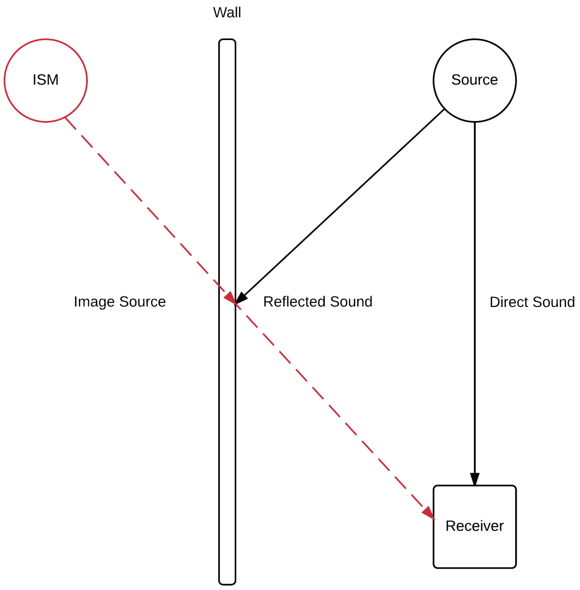


Figure 3: Illustration of the image source method (by the author) showing how the reflection from the sound source to the receiver is modelled by a secondary sound source.

Hybrid Method

Ideally, the ISM would be used to calculate all sound rays due to its accurate results, however due to the computational limitations, Odeon uses a hybrid method to provide a reasonable compromise between calculation time and accuracy.

The hybrid method first uses the ISM to calculate a number of image sources up until a specified reflection order determined by a Transition Order (TO). For example, if the TO = 2, the image source method will allow a ray to reflect twice which will produce a number of image sources at which point it will then switch to using the ray tracing method to calculate a statistical model of how the rest of the rays might interact with the room. This gives the user the choice between computation time and accuracy.

Scattering

Unless a surface is infinitely smooth (which almost all surfaces are not), when sound comes into contact with that surface it is scattered at a angle that deviates from that at which it hit the wall (the angle of incident). Odeon takes this phenomena into account by using 'Vector Based Scattering' [12]. This is done by adding the specular vector to a randomly reflecting vector that has been scaled by a value of 1-s, where s is the scattering coefficient applied to a surface (ranging from 0 - 1, covered in more detail in section Surface Types). This essentially calculates how much a ray should deviate from a specular reflection based on how 'rough' a surface is, where the rougher the surface, the higher the scattering coefficient. Figure 4 shows an annotated image taken from the Odeon manual illustrating this concept.

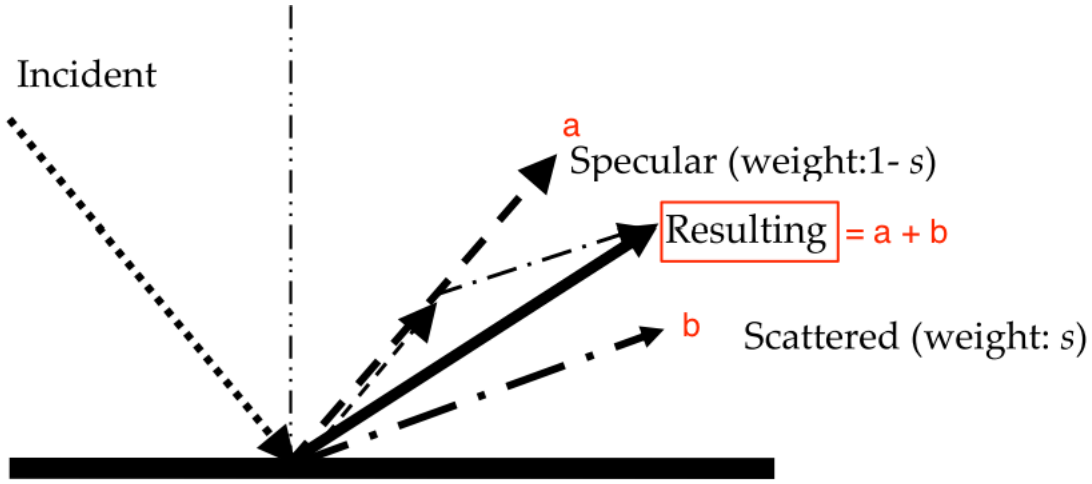


Figure 4: Annotated image from the Odeon manual [12] showing how scattered reflections are calculated, where a = the specular reflection, b = randomly scattered vector.

Room Modelling

In order to predict the acoustics of a space, Odeon requires a geometry file in the form of a .par file. This file contains information regarding the room dimensions, object dimensions and positions. This can be produced within Odeon itself by using the built in 'Extrusion Modeller' which allows a user to use a script like language to describe the rooms geometry. It is also possible to use 3rd party applications such as Google SketchUp [13], a software which is described in section Google SketchUp.

Material Selection

Selecting the appropriate absorption coefficients of the surfaces within a VAE is crucial to determining how sound will attenuate as it reflects around the room (as described in Ray-Tracing), thus the accuracy of the generated RIR. Odeon provides a material list with common materials with pre-determined absorption coefficients that can be assigned to the surfaces of a model read from a geometry file. This material list can be extended by creating new materials and assigning absorption coefficients to the appropriate frequency bands.

2.4.2 Google SketchUp

Google SketchUp [13] is an easy to use 3-D modelling software that can be used to produce room models. Unlike Odeon extrusion modeller, it allows the user the draw surfaces with a mouse, easily duplicate structures and provides simple measuring tools and markers to enable accurate modelling. Plug-ins such as SU2Odeon ("SketchUp to Odeon") [14] enables the user to convert the model into a .par file for Odeon to use as a geometry file.

2.4.3 Max/MSP

Max/MSP (Max) is a visual programming language that essentially provides pre-written blocks of code in the form of objects, providing the ability to chain them together in order to route and manipulate audio signals [15]. This allows user to quickly and easily produce audio applications (called patches) without having to deal with the complex side of audio programming. This is beneficial for students and researchers as it allows them to concentrate on the results rather than the application programming itself. Additionally, third-party plug-ins can be incorporated into a Max patch as libraries would in any other programming language. One third-party application utilised in this project is Spat. Developed at the research institute IRCAM [16], Spat is designed for real-time spatialisation of sound signals in Max without having to touch any code. It provides a simple way to convolve B-format audio signals with impulse responses as well as decode the B-format signals for any given number of speakers in any arrangement by simply providing the number of speakers and angles at which they are placed.

2.5 - Project Description

This section starts by explaining what the VSS is and how this project will build upon it. The motivation for doing so is covered and the project objectives and metrics are stated.

2.5.1 The Virtual Singing Studio

The VSS is a loudspeaker based room acoustics simulator used as a tool for analysing the correlation between room acoustic characteristics and vocal performance parameters as part of Dr Jude Breretons PhD Thesis [1]. It is comprised of a head-mounted microphone used to capture a real-time audio input from a singer, a software patch written in Max/MSP that convolves the audio signal with a number of Ambisonic B-Format RIR's using Spat and finally a spherical array of 16 loudspeakers for which the convolved audio signals are decoded and sent. This allows the performer standing within the speaker array to hear themselves as though they are stood in the VAE. In addition, a head-tracking device (an Oculus Rift [17]) is also used to track which direction

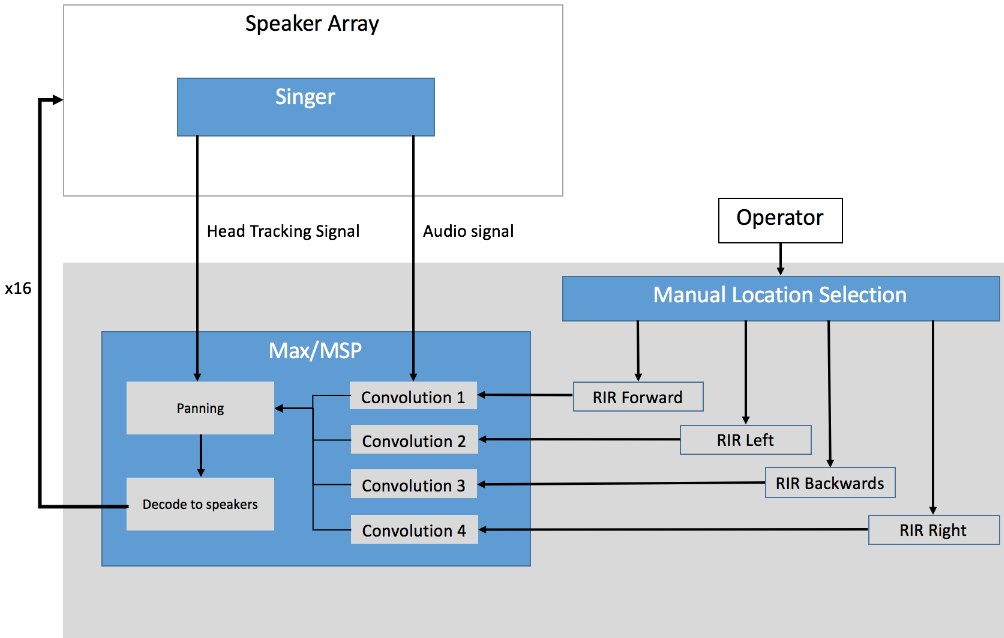


Figure 5: Flow diagram of the VSS showing how the audio signal and head tracking signal from the performer are used in Max before the convolved audio signal is sent back out to the speaker array.

the user is facing in the virtual space. A flow diagram of the system is shown in figure 5.

The VAE used in the VSS was initially the National Centre for Early Music, a space in York frequently used for musical performance. Using a Soundfield microphone, Ambisonic B-Format RIR's in each direction (front, left, back, right) were captured in 4 locations. The four directional RIR's are used to approximate the room acoustic phenomena that would occur if the user were projecting whatever sound they are producing in that direction. This is done by using the data from the head-tracking device to amplitude pan the convolved signals before sending them to the spherical speaker array.

2.5.2 Project Aims and Motivation

The VSS addressed a problem faced when trying to research how musicians perform in different acoustic environments: having to travel to each performance space with musicians and researchers. This also indirectly provides a solution for performers wanting to rehearse in spaces that are often inaccessible and would otherwise be expensive to book and travel to. By obtaining an RIR of the desired location, the only time travelling will be necessary is to initially obtain said RIR. However, one limitation of using the VSS is the restriction of position. If a performer wanted to try and sing at another point in the room, an RIR would have to be taken in that position too. This could be done initially, taking a range of RIR's in a number of positions, however it cannot be guaranteed that all positions desirable to the performer will be available. For this to be certain, technically an infinite number of RIR's would have to be taken. As this is obviously not possible, the next best thing would be to take a large grid of RIR's and simulate movement by interpolation between a number of neighbouring RIR's. This raises the question:

> How many RIR's are required to convince a user that they are able to move to any position they like within the virtual space?

As investigating this question would require producing a large grid of RIR's, measuring them in a real space would be impractical due to the sheer number that need to be taken for experimentation. Instead, Odeon can be used to produce a bulk of RIR's much quicker.

This technique has been considered previously in [18], referring to it as the *direct room impulse response rendering method*. However another method called *parametric room impulse response rendering* was favoured instead. This method actually synthesises RIR's in real time for a given position of the user within a VAE using geometric acoustic modelling methods. The reason this was chosen over the dynamic method was the fact that rendering the RIR for the users virtual position would be much more accurate and would not require storage space, as opposed to having to interpolate between a number of RIR that have to be stored somewhere. However, this method is much more complicated and limits itself to the accuracy of geometrical methods as wave-based methods are currently not able to run in real time (not even with using GPU technology, still taking approximately 10 minutes to render a single RIR [19]).

There are three reasons why using the direct rendering method in this project was initially considered:

1. Simplicity, as a real time RIR rendering system would not have to be produced.
2. The existing VSS system could be used as a starting point (thus adding to the simplicity).
3. The potential to use wave-based methods.

As using the parametric method requires the RIR's to be calculated in real time, it would not be possible with current technology to produce a more accurate RIR than can be obtained using the geometrical acoustic modelling methods. However, as the direct RIR method produces the RIR's offline, it could be possible to produce more accurate results. This has been investigated previously [20], where the RIR's calculated in Odeon are merged with the low-frequency accurate RIR's calculated using wave-based methods to produce an accurate full bandwidth RIR using calibration methods (a method to accurately mix the two together) proposed in [21]. However, early in the project it was decided that using wave-based modelling methods would be much too time consuming to implement along with carrying out the primary project aims given 1) the time scale of the project and 2) the time it takes to produce such RIR's compared to the ones produced in Odeon. It was therefore decided that a system would be built using only RIR's produced in Odeon which would still allow the question of how many RIR's are required to produce such a system to be investigated.

In order for the user to move themselves around the VAE, the Max patch that the VSS was built on had to be extended in the following ways:

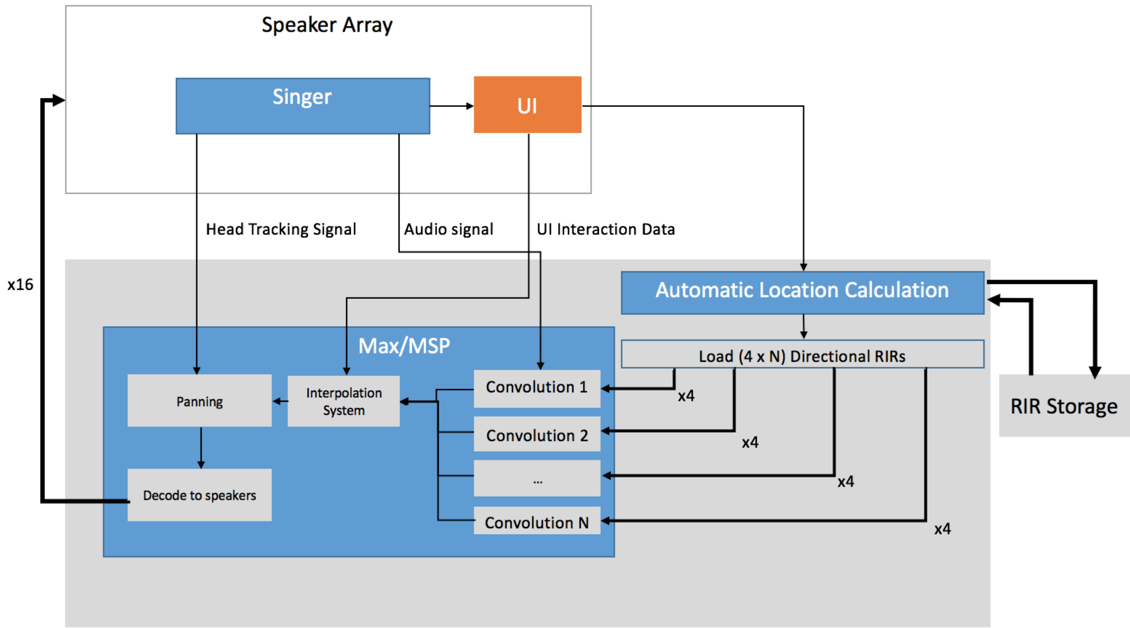


Figure 6: Flow diagram of the way in which the desired extended version VSS works

1. The production of a user interface that can be used remotely from within the spherical speaker array.

2. The extension of the Max patch that could accommodate the user interface and load the appropriate RIR files required to place the user in the desired location within the VAE.

3. The production of a system that can interpolate between the appropriate RIR positions.

Figure 6 shows an adapted flow diagram from figure 5 with the added user interface, automatic file searching functionality and interpolation system.

As the original implementation of the VSS uses real RIR's to simulate a VAE and the newly proposed implementation was to use synthetic RIR's, the difference in perception regarding how a user feels they are moving around the space was also to be investigated. This would provide an insight as to whether the newly implemented functionality of movement was worth sacrificing the accuracy of the real RIR's. For this to be investigated, real RIR measurements must be taken, therefore, the VAE to be modelled must be a building that was accessible for real RIR measurements.

To conclude, the project aims were:

1. To implement a direct RIR rendering system as an extension of the VSS
2. To investigate the number of RIR's that are required to convince a user that they can freely move themselves around a virtual space without limitation
3. To investigate the difference in the perception of mobility in the virtual space when using

real RIR's and synthetic RIR's

2.5.3 Project Objectives

Given the project aims and the background provided, the following is a list of objectives that were set in order to complete this project:

1. Find an appropriate room to be modelled as a VAE
2. Digitally model the room using in Google SketchUp
3. Import room model into Odeon to finish material selection and RIR settings
4. Produce a grid of RIR's that can be used to interpolate between to simulate user positions
5. Record real RIR's in locations that can be compared to synthetic RIR's
6. Extend upon existing software patch to accommodate new functionality
7. Preform user tests:
   - Test #1: Does the perception of distance change when using real or synthetic RIR's?
   - Test #2: How far does the user have to move in the given VAE before they notice they have moved
   - Test #3: How many RIR's are required for interpolation for the user to feel they are moving around the space freely

IMPLEMENTATION

This section describes the practical work undertaken as part of this project, explaining the choices made and issues raised.

3.1 - ROOM CHOICE

The first step required before any progress could be made was to find a suitable room for modelling and recording RIR's. The following room features were kept in mind when searching for a room to use as part of this project:

- Size: Large enough to be used as a singing space.
- Simplicity: Simple enough architecture to be able to model accurately with the time available.
- Accessibility: The room had to be easily accessible for taking multiple measurements and RIR recordings.

Hendrix Hall is a large lecture theatre on the University of York campus and fulfilled the above stated requirements, this was chosen to be used in this project. The room contains retractable seating leaving the large space in the centre open and other than two desks at the front of the room, unobstructed. The room is architecturally simple being almost perfectly rectangular with the occasional wall indent. With it being located on the universities campus it could be booked for any time of the week meaning it could be accessed when necessary and for free.

3.2 - USER TEST PLANNING

In order to perform the final user tests (test #1 and test #2), a minimum distance between RIR's had to be decided. Upon finding and measuring Hendrix Hall, the maximum number of RIR's that would need to be produced could be calculated for a given minimum distance of RIR location separation. Originally a separation of 0.5m was considered, however, upon obtaining the dimensions of Hendrix Hall, it was calculated that this would involve producing 1,920 RIRs. As it was not known how long producing a bulk of RIR's would take, it was decided that a 1m separation between RIR's would be sufficient, meaning a total of 960 RIR's would need to be calculated.

As user test #3 would require producing a number RIR grids of different densities, the **maximum** distance between RIR's that could be used was calculated. It was decided that a maximum distance for RIR separation would be 5m, which would produce a 3x3 grid of RIR's. This meant that a total of 5 grids were to be used in user test #3, each containing RIR's separated by an extra meter in each from 1m to 5m.

3.3 - Real RIR Recordings

For testing the difference in perception of movement when using synthetic or real RIR's, real measurements were taken in Hendrix Hall.

3.3.1 RIR Measurement Set Up

For the VSS it is desirable to obtain RIR's that can be used to represent the topology of a singer, i.e mouth (sound source) below the ears (receiver). To achieve this, a Genelec 8040B [22] loudspeaker was used as a directional sound source and a Soundfield St450 MKII microphone [23] was used as the receiver to record the three dimensional sound field in Ambisonic B-format, as is required for them to be used in the VSS.

Figure 7 shows an image of an approximate human head topology sound source and receiver set up. The Genelec is placed 1m above the ground (from the base of the speaker to the floor) and the Soundfield microphone placed 0.6m above the sound source. Ideally the receiver would be placed closer to the sound source to more accurately represent the distance between the ears and the mouth, however due to the physical dimensions of the equipment being used, this was not possible. The sound source was placed 1m off the ground simply due to the limitations set by the maximum height of the microphone stand, otherwise a more realistic average hight for a human would have been used.

The overall set up used for recording the RIR's can be seen in figure 8, showing: **(1)** the Genelec and Soundfield microphone in the above mentioned source and receiver set up with a special soundfield cable running into **(2)**, the soundfield unit used to output an Ambisonic B-format signal through four XLR-to-jack cables to **(3)** a Fireface UXC audio interface plugged into **(4)** a Mac running the digital audio workstation Reaper.

Reaper was used to:

1. Record directly to a single 4 channel track where channels 1 - 4 recorded the W, X, Y, Z channels of the soundfield microphone respectively.
2. Output a 15 second long sinusoidal sweep to the Genelec.

Using reaper to do both simultaneously avoided synchronisation issues often faced when using separate devices to output and record a signal.

3.3.2 Positions

Four positions within the room were chosen and marked with tape for the RIR positions shown in figure 9 where:

Position	(1)	(2)	(3)	(4)
Coordinates [x(m), y(m)]	[9,9]	[4.5,9]	[2,9]	[10.13,1.46]

For each position, an RIR was taken in four different directions. Starting at 0° , the sound source

Figure 7: Human head topology RIR measurement set up with a Genelec 8040B sound source placed 1m off the floor, 0.6m below a Soundfield microphone used as a receiver.

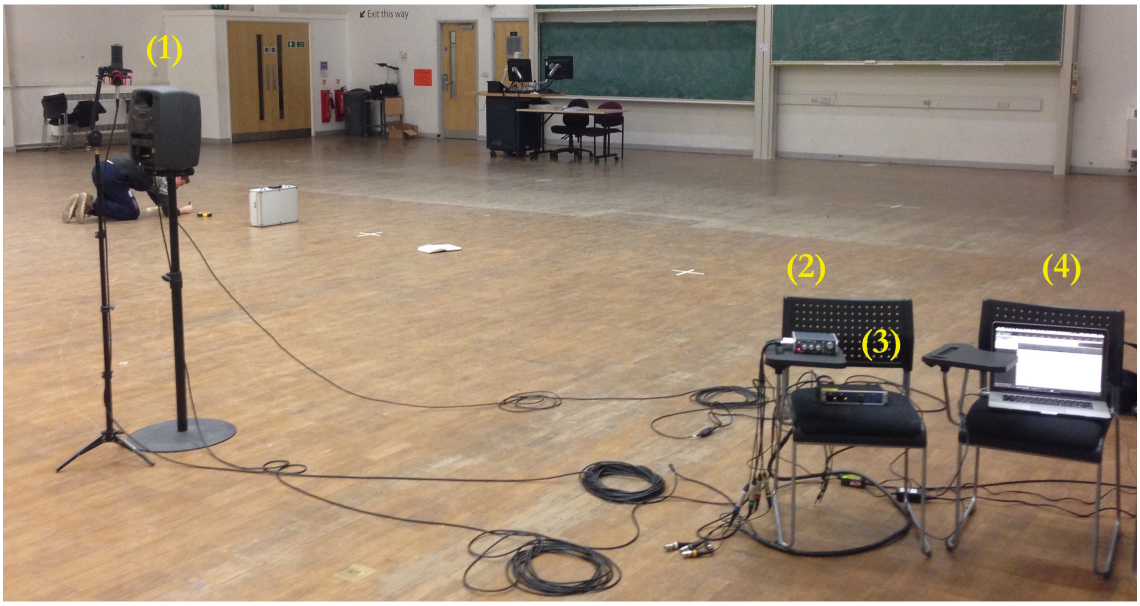


Figure 8: Real RIR measurement set up. **(1)**: Sound source and receiver set up **(2)**: Soundfield interface **(3)**: Fireface UXC audio interface **(4)**: Mac running Reaper to output sinusoidal sweep and record B-format input.

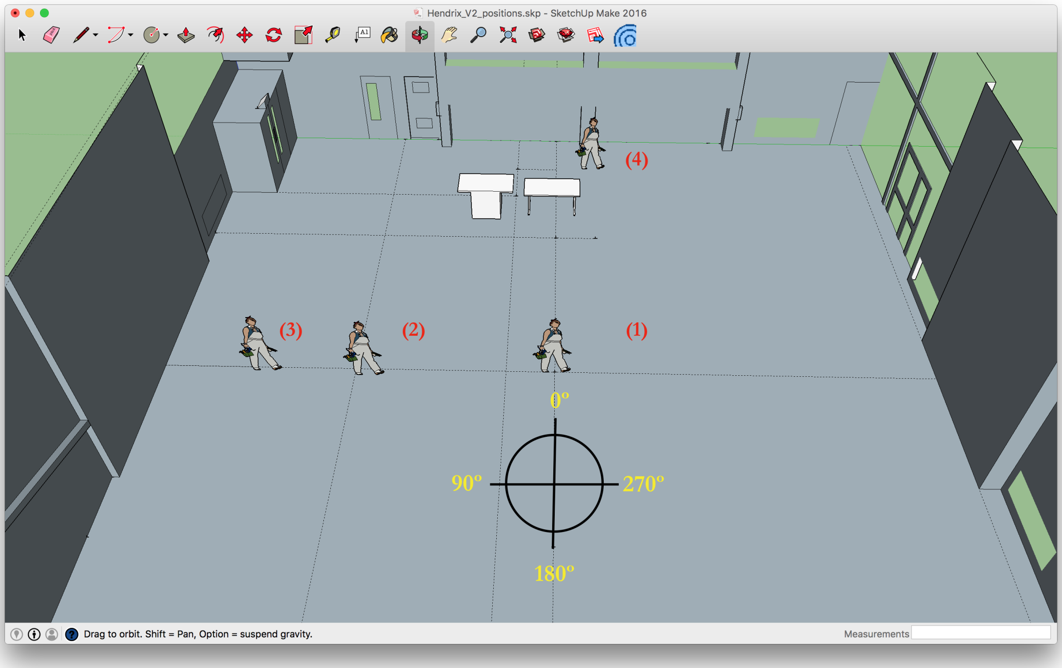


Figure 9: Google SketchUp model showing the positions of where the real RIR's were taken. The compass shows 0°facing what has been defined as the front of the room.

was rotated anti-clockwise in 90° increments (anti-clockwise rotation is a standard in Ambisonics). Rotating the sound source and keeping the receiver at 0° (facing the blackboards) ensures that when the user turns their head in the VSS the sound field does not rotate as they do. Once the four directional RIR's were recorded, the source and receiver were moved to the next marked location with care being taken to make sure the receiver is placed the same distance away from the sound source each time.

3.3.3 Measurements

The sound source signal used was a 15 second long exponentially swept sinusoid ranging from 20Hz to 20kHz with an 8 second padding to ensure that the room could be vacated before the sweep began. The sweep was produced using the Matlab function `generatesweep.m` taken from the departmental website [24] which produced both the sinusoidal sweep and an inverse sinusoidal sweep used for the deconvolution process required to produce an RIR, which was done at a later stage.

3.3.4 Measurements Post-Production

Once the measurements had been recorded, the first 8 seconds of the files were muted in order to remove noise such as footsteps and the slamming of doors as the room was vacated. Then the signals were deconvolved with an inverse of the sinusoidal sweep to time align the frequency dependant room reflections. This was done using the deconvolution Matlab function `deconvolve.m` taken from the departmental website [24].

3.3.5 Issues

Several issues arose when taking the initial RIR's. Simply connecting the outputs from the Soundfield converter to the incorrect inputs on the Fireface audio interface meant that the initial recordings contained tracks that were not necessarily recorded in the correct order, i.e the tracks were not recorded as [W, X, Y, Z] but could have been recorded in a possible 24 combinations. This meant that when the RIR's produced using the recorded sine sweeps were convolved with an audio source, the localisation of the sound source would be incorrect.

In an attempt to salvage the potentially ruined RIR recordings, the tracks were analysed through observation and convolution with test tracks. However, though it was possible to narrow down which order the channels might have been recorded in, pinpointing the track order could not be done with 100% accuracy. Therefore, Hendrix Hall was rebooked and the measurements were taken again.

3.4 - Room Modelling

3.4.1 Room Measurements

Before taking room measurements, a quick top-down map of the room was made, highlighting objects, wall indents or protrusions as well as where doors and windows existed. The dimensions of the room were then measured in meters and noted on the not to scale room map. The tool used for taking measurements was a DeWalt laser distance measurer [25], allowing for accurate measurements of distances that would otherwise not be accessible, such as the distance between the lights on the roof. Once the basic layout had been mapped, more detailed diagrams of individual walls were made, paying close attention to window and door dimensions, distances between doors and window panes, the depth of the radiators on the walls etc, as these small variations in surface depth would greatly influence how sound would reflect around the room. An example of an annotated blueprint can be seen in figure 68 in Appendix A.

3.4.2 Designing the room

The blue prints were used to model the room in Google SketchUp, starting with a hollow rectangle with the dimensions of Hendrix Hall (18.3m x 18.2m x 5.5m). The wall indents and protrusions were then modelled by using a push/pull tool. Figure 10 shows an early iteration of the SketchUp model where several wall protrusions have been modelled.

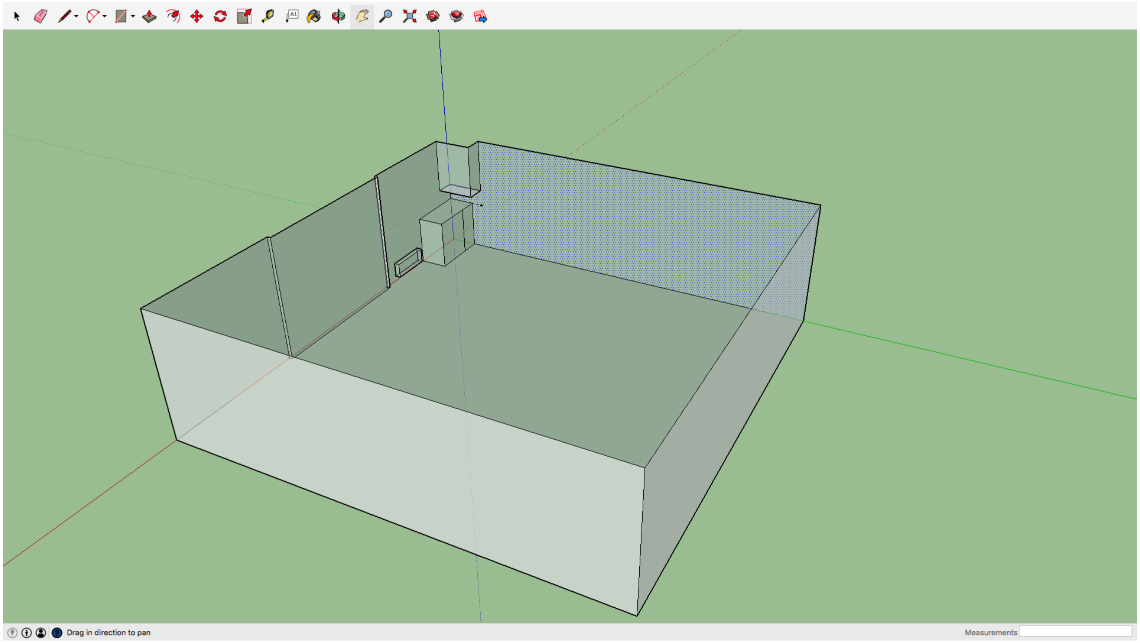


Figure 10: Early iteration of the Hendrix Hall SketchUp model with a few early wall protrusions being modelled such as the entrance door, a radiator and one of the wall indents.

Hendrix Hall predominantly consists of flat surfaces, though some more complex surfaces include:

- Lights (concave curves)
- Canvas roof hangings (convex curves)
- Projector hangings (poles)
- Radiator, door and vent grills

These objects were initially designed to look as close to the real thing as possible, however the objects with curved edges posed a problem when importing the model into Odeon. This is because SketchUp uses a large number of short surfaces to represent curves, as can be seen on the left of figure 11 which shows the initial models of the lights, projectors and roof hangings. According to the Odeon manual [12], when modelling a room for acoustic simulation purposes, it is more accurate and less time consuming to keep the model simple and to add the appropriate scattering coefficients or materials in Odeon itself. This also applied to the objects (such as the radiators) that contained a grilled surface, as a specific 'grill' material could be selected from Odeons material list (see section Material Selection). The objects were redesigned to Odeons specifications by using simple geometry, where the roof hangings were represented as four joint slanting surfaces and the lights and projector poles were represented as simple rectangles, as can be seen on the right in figure 11

Images of the final model compared to the real room can be found in Appendix A, in figures 69, 70 and 71, showing that the room was carefully recreated in hope of improving the accuracy of

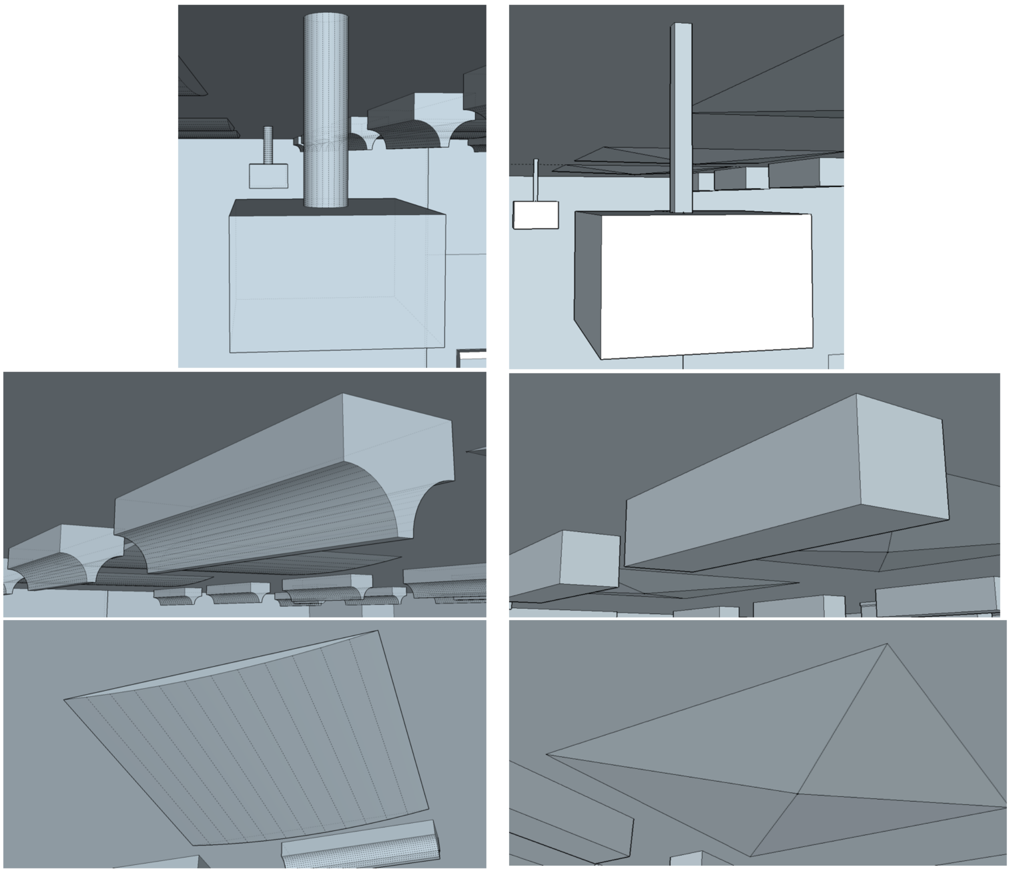


Figure 11: Comparison of the original complex models (left) and the their simplified versions (right)

the results obtained from Odeon. The Google Sketchup file is also available in folder 4.1 as part of the supporting material.

3.5 - ODEON

The Odeon project described in this section can be found in folder 4.2.1.

3.5.1 Water Tightness Test

Once the SketchUp model had been exported as a .par file using the SU2Odeon plug-in [14] it could be opened in Odeon and checked to ensure that there were no gaps in the model for which rays to escape. If this was the case, the model would have to be fixed in SketchUp and reimported into Odeon. Odeon makes checking the model easy by running a '**water tightness**' check where a large number of rays are traced around the room, seeing whether any of them manage to escape. Figure 12 shows the Hendrix Hall model undergoing such a test. Once it was ensured that the model was fit for use, the surface materials could be assigned.

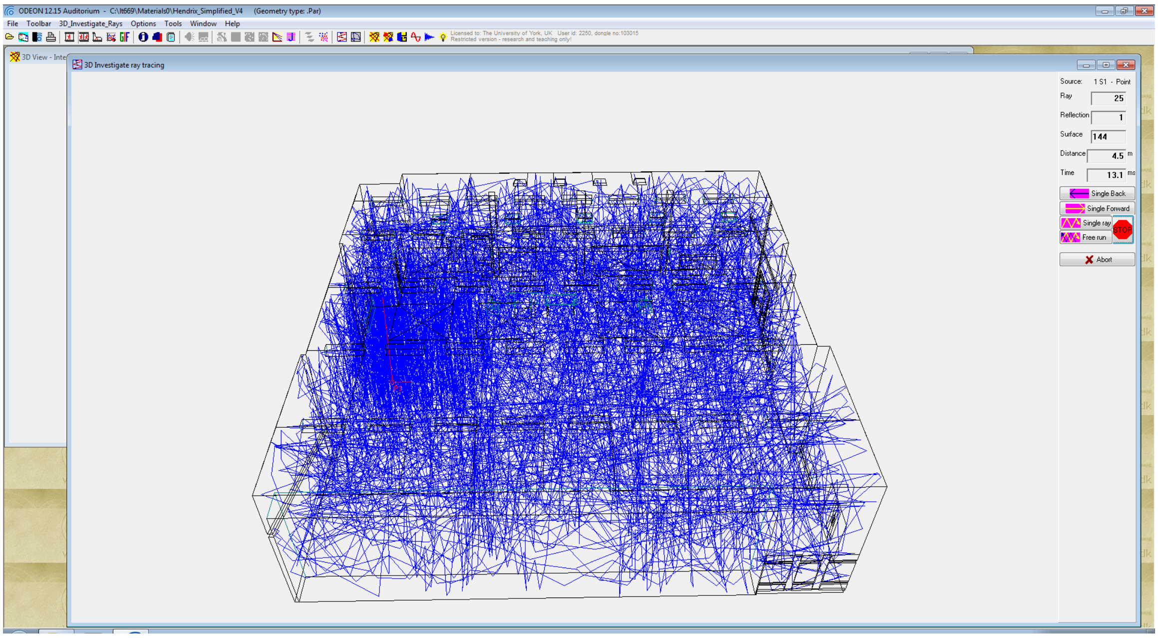


Figure 12: Hendrix Hall model undergoing a water tightness test in Odeon.

3.5.2 Material Selection

Surface materials greatly influence how sound reflects around a room, effecting reverberation time and the rate at which frequency bands are absorbed. It is therefore imperative to assign materials that closely match those within the real room in order to produce an accurate representation of the acoustic environment. For this, Odeon provides a list of common materials often found when constructing buildings.

Initial Materials

Odeon's material list appears to be designed for the auralisation of structures with very basic interior. Due to a lack of choice, exact materials in the room could not be modelled accurately, however, the closest match to what was thought to be the true material were selected as an approximation. In some cases, appropriate replacement materials were not available, therefore new materials had to be added to the material list. This can be done by finding a materials absorption coefficients and selecting '**Edit an existing material**' where a new materials absorption coefficients can be entered and saved, as shown in figure 13. Absorption coefficient values range from 0 - 1 (0% - 100%), indicating the percentage of attenuation applied to the selected frequency band upon a contact with the surface. Required materials for which an appropriate replacement could not be found in the material list are listed in table 2.

Material	Surface applied to
Hard Plastic [26]	Roof lights and projector covers
Mineral fibre [27]	Ceiling Tiles
Slate [1] [28]	Blackboard

Table 1: Table of materials for which absorption coefficients were sought and added to Odeon's material list.

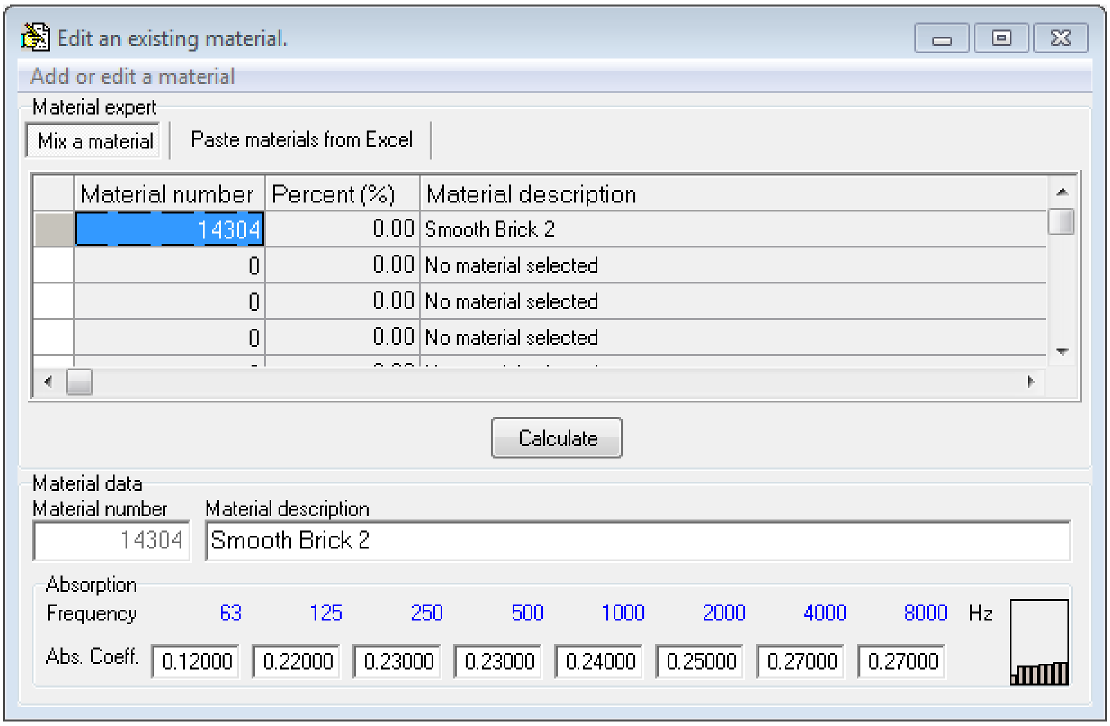


Figure 13: Absorption coefficient editing window in Odeon used to add unavailable materials.

Surface Types

For a number of surfaces it is appropriate to edit properties other than just their absorption coefficients. As explained in section Designing the room, the roof hangings are constructed of four joint slanting surfaces. By changing the individual surface types from **'Normal'** to **'Fractional'**, Odeon avoids erroneously calculating diffraction due to each of the individual surfaces and treats them as a whole surface. For surfaces such as the contracted seating shown in figure 14 where gaps are present in the overall structure, it was possible to model this as one solid object and to set a **transparency** value. A transparency value of 0 means the surface is a solid, whereas a value of 1 makes a surface totally transparent, allowing rays pass to straight through it. A transparency value of 0.3 was chosen as a reasonable estimate, the effects of which are shown in figure 15, showing how

[1]The absorption coefficients provided were only available for the 125Hz to 4kHz octave bands, therefore absorption coefficients for the 63Hz and 8kHz octave bands were given a value of 0.1

a ray can pass through the front of the object, reflect around the inside and eventually reflect back out again.

Figure 14: Contracted seating in Hendrix Hall showing gaps in the structure.

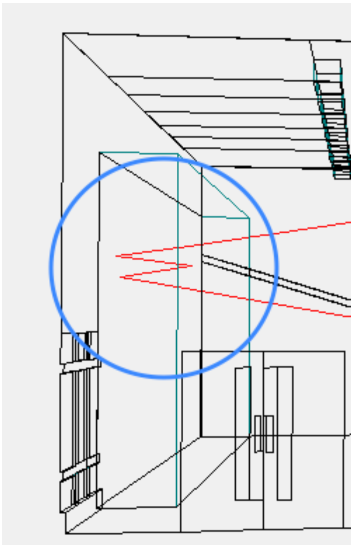


Figure 15: Blue circle highlights a ray penetrating the modelled seating area, reflecting 3 times and escaping the seating area due to the transparency value set.

As previously mentioned in section Designing the room, the lights were modelled as simple rectangles as opposed to the more complex objects made from a large number of surfaces with the intention of more accurately modelling the scattering effect (described in section Scattering) due to their shape by altering the objects **Scattering Coefficient**. Odeon provides a table of initial indicators for possible scattering coefficients, shown in figure 16 which can be used to select the scattering coefficient for a surface that fits a similar description to the one given. Given the vague descriptions, a scattering coefficient of 0.2 was selected for the lights.

Material	Scattering coefficient at mid-frequency (707 Hz)
Audience area	0.6–0.7
Rough building structures, 0.3–0.5 m deep	0.4–0.5
Bookshelf, with some books	0.3
Brickwork with open joints	0.1–0.2
Brickwork, filled joints but not plastered	0.05–0.1
Smooth surfaces, general	0.02–0.05
Smooth painted concrete	0.005–0.02

Figure 16: A table provided in the Odeon Manual [12] that can be used to set an approximate scattering coefficient for surfaces similar to those described.

RIR Comparison

Once the initial materials had been selected, the authenticity of the RIR's were checked by comparing their frequency content against that of a real measured RIR from Hendrix Hall (see section Additional Functionality). This was done by using Matlab to plot the spectrograms of the omni-directional W channel of both RIR's, as can be seen in figure 17. The spectrogram shows the high frequency content in the real RIR attenuating in a smooth roll off fashion from 20kHz to about 8kHz starting from approximately 0.2 seconds in, whereas the full audible spectrum appears to attenuate almost evenly in the RIR produced in Odeon.

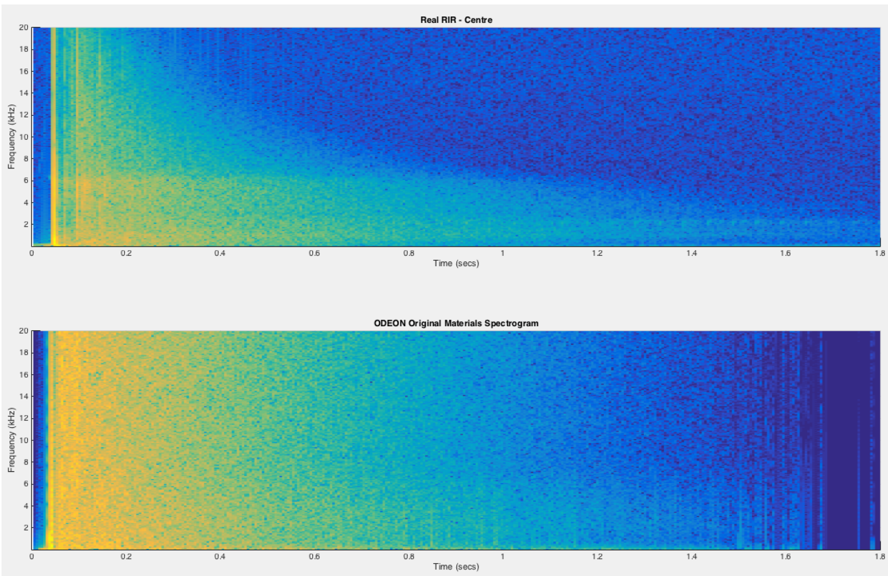


Figure 17: Spectrograms of real RIR (top) against spectrogram of first rendered RIR from Odeon.

As the most obvious difference between the two is the difference in high frequency attenuation, large surfaces with low valued high frequency absorption coefficients, such as the walls and ceiling, were edited to absorb more of the high frequency content. Several iterations can be seen in figure 18, showing the spectrogram of the real RIR and several iterations of the RIR's rendered from Odeon. The Spectrogram at the bottom is that of an RIR produced in Odeon with the final material selections. The difference between the real RIR and the final Odeon RIR can be seen more clearly in figure 19. This was done by adding 10% onto the existing absorption coefficients for the 63Hz - 2000Hz octave bands, 20% to the 4000Hz band and 30% to the 8000Hz band. This also included editing the ceiling material coefficients by adding an extra 10% to the 8000Hz octave band.

Table 2 contains a list of audio samples (can be found here or in file 1.4) indicating the edits made to the material list in order to produce each one, with their corresponding spectrogram graph name which can be found in figure 18.

Audio file	Graph Name	Absorption Coefficient Edits		
		Material	**Addition**	**Octave band**
RealRIR.wav	Real RIR	NA		
OdeonOriginal.wav	ODEON Original	NA		
Odeon3RIR.wav	ODEON 3	Ceiling (Mineral Fibre)	+10%	8kHz
Odeon6RIR.wav	ODEON 6	Walls (Solid Brick)	+10%	All
Odeon7RIR.wav	ODEON 7	Walls (Solid Brick)	+10%	4kHz
			+20%	8kHz
		Ceiling (Mineral Fibre)	+10%	8kHz

Table 2: List of audio samples with corresponding spectrogram title in figure 18 and the edits made for each one. (Only Odeon(3)(6)(7) audio sample are shown here as the other audio samples produced as a result of other material list iterations provided no relevant information).

By analysing figure 19 it can be seen that it was possible to produce a much more fitting RIR compared to the one produced using the original materials. By listening to the audio samples, it is clear that though the final Odeon RIR (Odeon7RIR.wav) is much more similar to the real RIR (RealRIR.wav) than the one produced using the original surface materials (OdeonOriginal.wav) in terms of reverberation time, they still sound very different in terms of their frequency content, where the RIR's produced in Odeon lack what the author considers *depth*. This is most likely due to the fact that the geometrical acoustic modelling methods used to produce the RIR do not accurately reproduce the low-frequency content, the reasons for which were discussed in Odeon: Simulating Room Acoustics. However, it can be said that the final Odeon RIR sounds much more similar to the real one than the other three RIR's produced using earlier iterations of the material list.

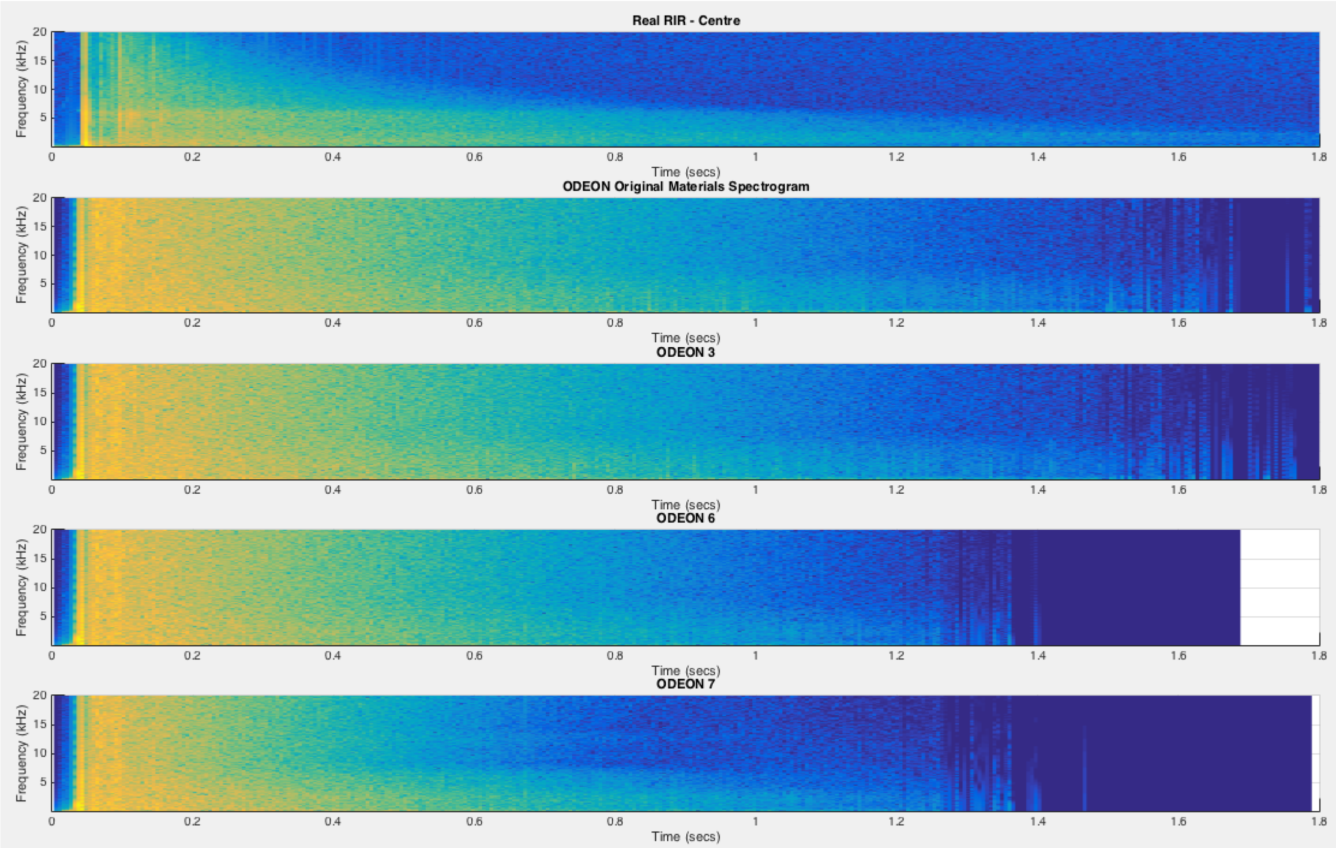


Figure 18: Spectrogram of real RIR (top) against the spectrograms of several rendered RIR from Odeon with different material absorption coefficients where ODEON (bottom) shows the final RIR used. Spectrogram titles and corresponding material edits can be found in table 2

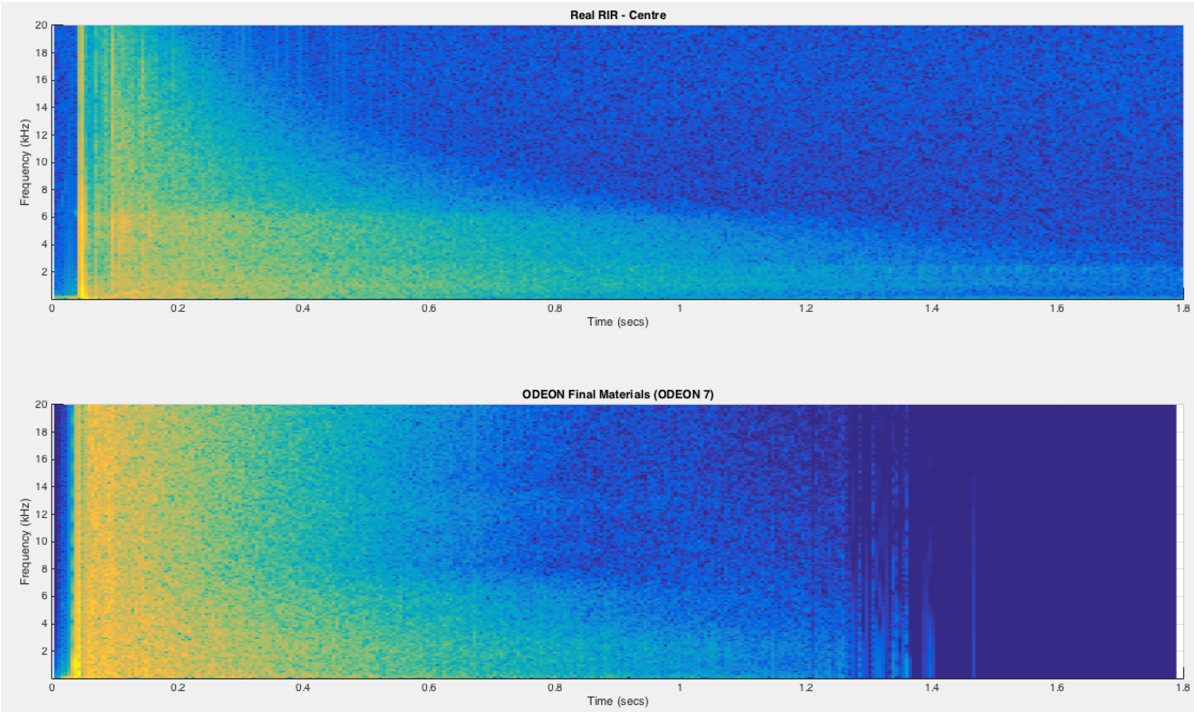


Figure 19: Spectrograms of real RIR (top) against Odeon RIR from Odeon with final material absorption coefficients showing a more convincing replica of the real RIR by editing the rooms material absorption coefficients.

The full material list is available in Material_List_V2.xlsx, showing which materials are applied to each surface within the Odeon model, as well as a list of absorption coefficients iterations that were applied to the brick walls and ceiling can be found in Absorption_Coefficients.xlsx. Both can also be found in file 4.3.

It was later discovered (see section RIR Topology Problems) that the RIR's used to decide the final materials were incorrect, a result of which caused them to be much louder than they should have been. To ensure that the final surface materials that were selected when using the incorrect RIR's were still appropriate for the new, quieter ones, two new RIR's were rendered using both the original materials and the final materials. Both were plotted to compare against the same real RIR used before, the results of which are shown in figure 20. As the new Odeon RIR's were so much quieter, the real RIR was reduced in level to match those produced by Odeon. As it can be seen, the final materials used in the room model reduce the length of the reverb and attenuate high frequencies quicker than when the room model contained the original materials as they had shown previously.

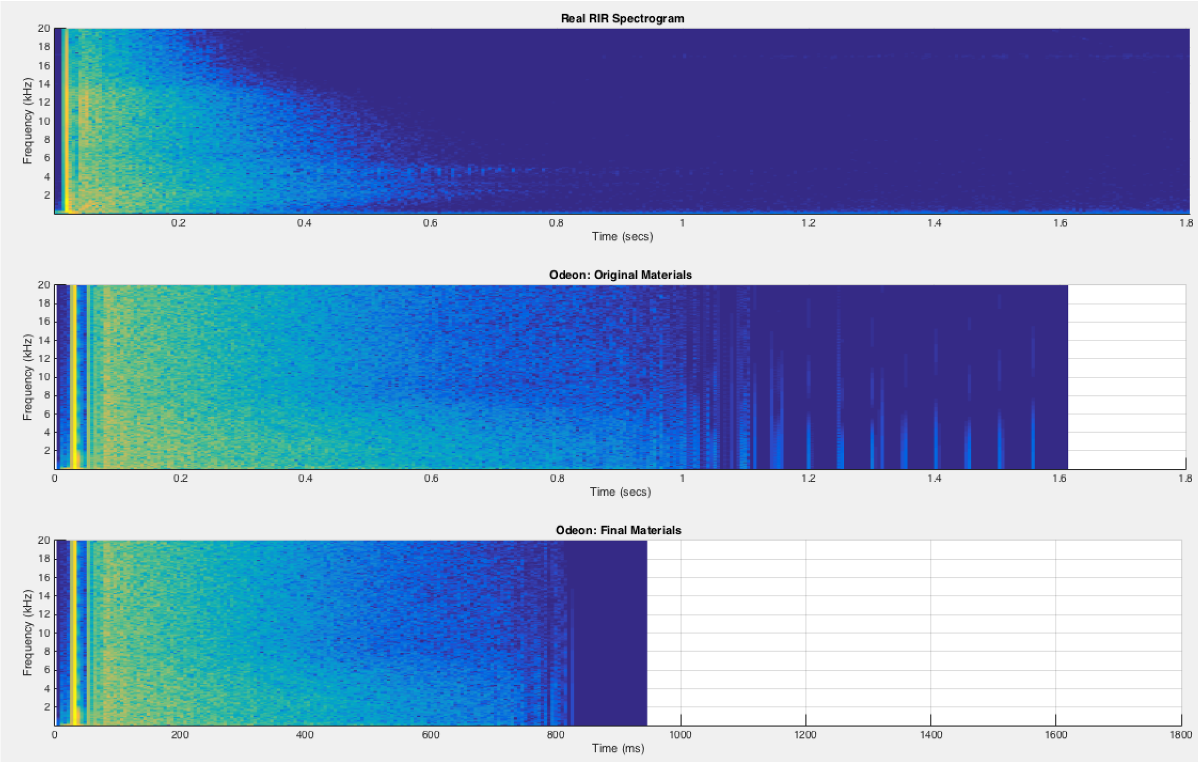


Figure 20: Spectrograms of correct Odeon RIR's with original materials (top) and final materials (centre) to compare against real RIR that has been level calibrated (bottom).

3.5.3 Odeon Output Settings

Ray Settings

Odeon provides the following options for RIR rendering:

Astop - The maximum possible attenuation of each octave band.

Apass - Ripple of octave band filters in dB.

Band overlap - 100% overlap gives a smooth transition between the FIR octave band filters as they are not completely rectangular.

Maximum reflection order - Maximum allowed is 2000 as is the default. This means a ray can only reflect around a room 2000 times before the simulation is stopped. This prevents trapped rays from prolonging the impulse response if it never reaches the receiver.

Late Rays - These are emitted from the source and reflected according to the Vector Based Scattering Method (VBS) (described in section Scattering) taking into account scattering due to surface size and roughness.

Transition Order - As explained in section Odeon: Simulating Room Acoustics, a TO can be set to determine the number of early rays sent out to find a number of wall combinations for reflections

Audio Sample	TO	LR’s	Graph Name
TO_2_LR_1285.wav	2	1285	TO_2_LR_1285
TO_2_LR_20560.wav	2	20560	TO_2_LR_20560
TO_4_LR_1285.wav	4	1285	TO_4_LR_1285
TO_4_LR_20560.wav	4	20560	TO_4_LR_20560

Table 3: Table of audio samples with the corresponding settings information and graph name. Samples can be listened to here or in file 1.3

using the ISM. After this number of rays has been reached, the ray-tracing method is used.

All settings apart from Late Rays (LR’s) and TO are left at their default values as they are sufficient for accuracy and prevent the increase of computation time which will be needed given the large number of RIRs required for free movement.

Odeon suggests two possible modes, **Engineer** and **Precision** which both suggest using a different number of LR’s which are 1285 and 20560 respectively. As both the TO and LR’s value increase both the accuracy and computation time of the RIR’s, different combinations of the values were tried and RIR’s were rendered for comparison. Figure 21 shows plots of the rendered RIR’s with the name indicating the TO and LR’s value. For example, ‘TO_2_LR_1285’ indicates the sample was produced where TO = 2 and LR’s = 1285. Table 3 can be used to clarify the settings used for each RIR in figure 21 and indicates which audio sample can be listened to. For consistency, all RIR’s were taken from the centre of the modelled room.

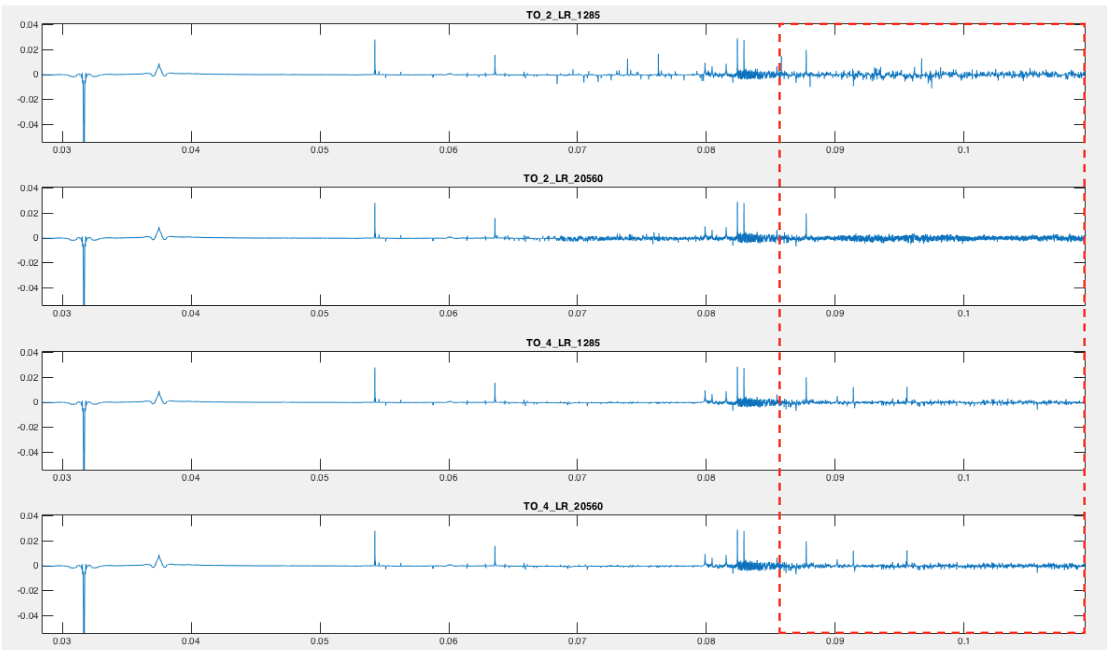


Figure 21: RIR’s rendered using different TO and late ray values

Through aural analysis, it can be heard that the number of LR’s used has an audible impact, where

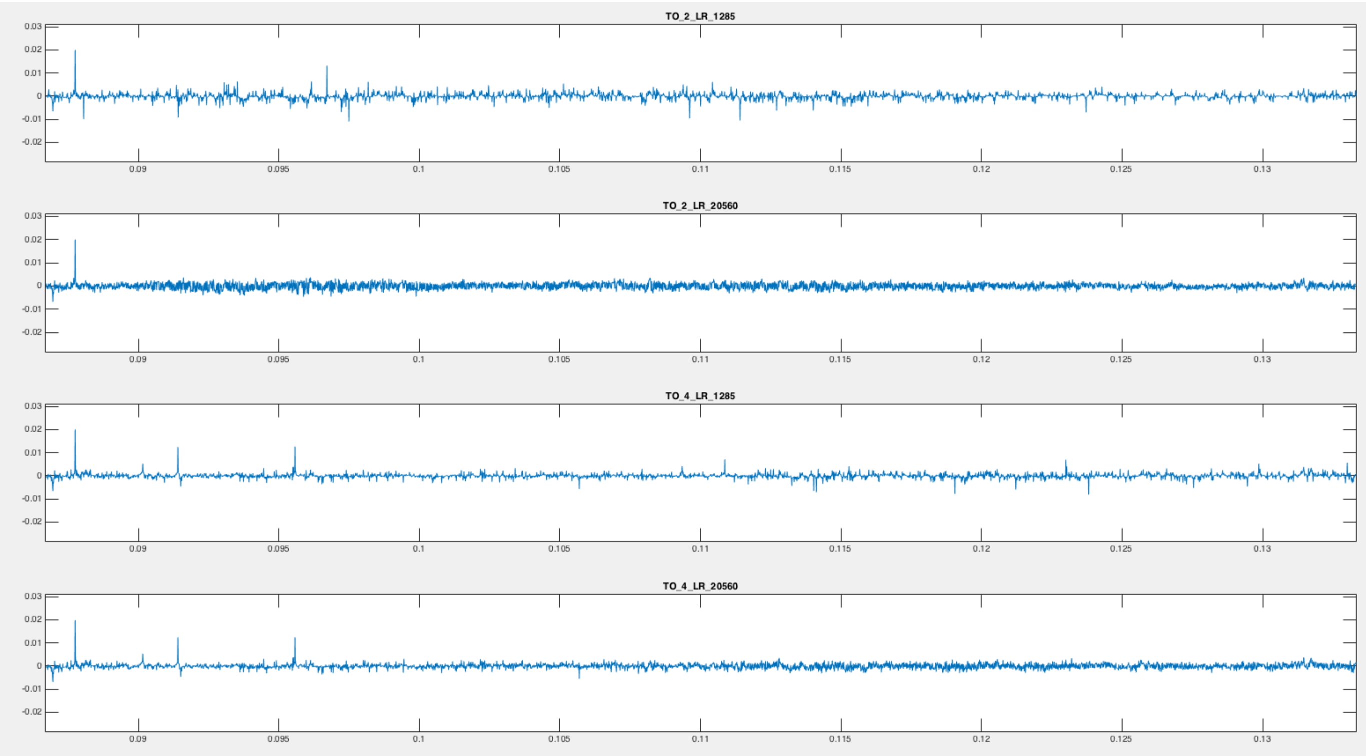


Figure 22: Zoomed in section indicated by the red dashed line in figure 21

a lower LR's value produces an RIR that sounds *grainy*. This can be seen in figure 22, which is zoomed in on the section highlighted in red in figure 21 showing the RIR's from approximately 0.085s, where the difference caused by the different setting combinations can be seen. The two RIR's with a high LR's value (2nd and 4th plot) show more densely packed reflections due to the fact that more rays are used. This produces a 'smoother' more natural sounding reverb tail.

Figure 23 shows the same RIR's from the start of the plots. For the RIR's produced with a TO value of 2 (1st and 2nd plots), it can be seen that there are a lot more randomly occurring peaks than in the RIR's with a TO value of 4. It is assumed that this is caused by the fact that the ray-tracing method is used much earlier on in the top two RIR plots, thus the randomly reflected rays cause more randomly occurring peaks. Though the difference in using a TO of 2 or 4 can be seen, it is not audible by listening to the RIR audio samples.

The time difference taken to render the RIR's is also an important factor when choosing which of these settings to use. Given the potentially large number of RIR's required, a slightly more accurate output may result in a much greater calculation time. Odeon uses information gathered when previously rendering RIR's to speed up calculations for other RIR's being rendered in the same room, however there initial time taken to calculate each of the RIR's was noted and compared for reference.

Initially it was found that increasing the TO value from 2 to 4 increased the time taken to render by approximately 20 seconds, at which point it was decided that a TO value of 2 would be used,

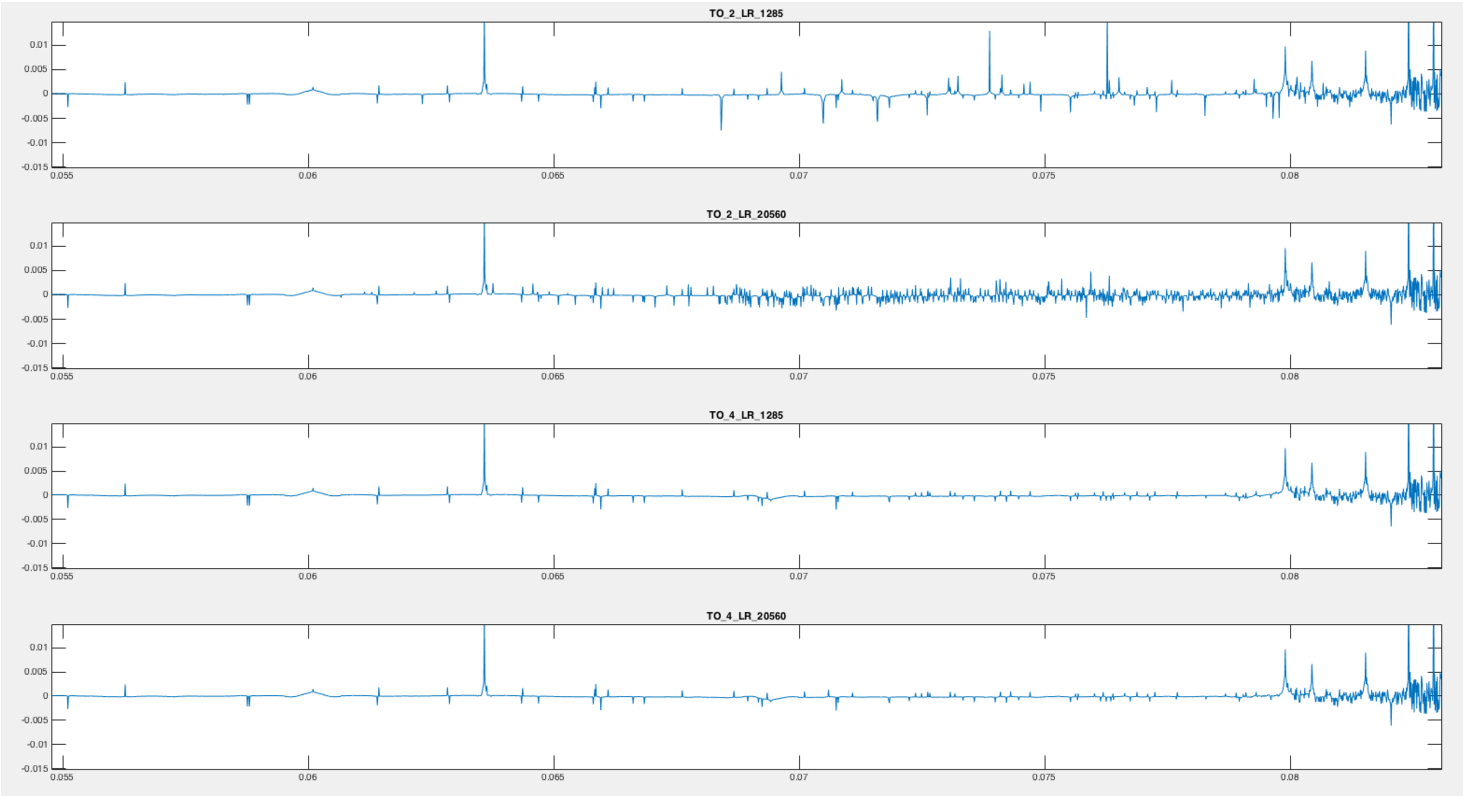


Figure 23: Zoomed in section showing the effects of using a high TO value

given that it made no perceptual difference. However, much later it was realised that inaccurate calculation times may have been given due to the fact the project was being stored on the University network, thus adding variable delay times. The project was later stored on a local C drive, eliminating any network speed issues and inconsistencies. After doing so, the difference in TO values made almost no difference to the calculation times. In hindsight, a higher TO value could have been used to produce more accurate RIR's.

It was found that the LR's value made quite a difference to the calculation time:

Transition Order	Late Ray Value	Calculation Time (s)
2	1285	11
2	20560	28
4	1285	11
4	20560	28

Despite the increase in calculation time by 17 seconds, the inaccuracy of RIR's produced when using the lower LR's makes using the higher value justifiable.

As the VSS requires B-format audio files, the B-format option was selected in Odeons options menu.

Directivity

Without selecting otherwise, Odeon uses an omni-directional point source as the sound source. This can be changed, however, by providing a .cf2 file. This file stores loudspeaker performance data and polar plots in what is known as a Common Loudspeaker Format, a standard used by loudspeaker manufacturers. By importing one of these files, Odeon simulates the directionality of the selected loudspeaker as the sound source. This can be used to attempt to more accurately recreate an RIR that would be taken in a real space, by finding the .cf2 file that corresponds to the loudspeaker used for the measurement.

It is also possible however, to input a custom directivity pattern to model a specific sound source. This provides an opportunity to create a more accurate sound source for a human head, something desirable when creating RIR's that will be convolved with the audio input from a singer. Techniques for recording the directivity of the human head have been reviewed and improved upon in [29], however only directivity data for the horizontal plane was recorded. Calum Armstrong, a student from the University of York has recently investigated the directivity of a human head in both the vertical and horizontal planes ranging from the 63Hz to 8kHz octave band [30] and generously provided said data to be used in this project. Figure 24 shows the directivity pattern editor window in Odeon with the horizontal and vertical directivity patterns for the 8Khz octave band taken from [30].

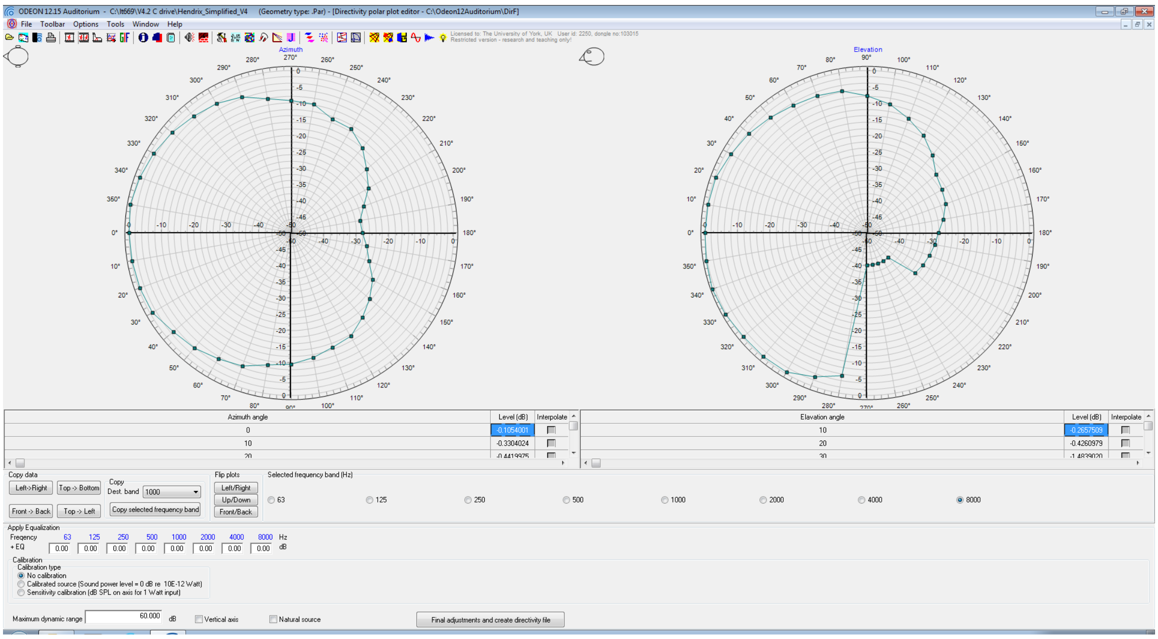


Figure 24: Directivity pattern editor window in Odeon showing the directivity pattern for the horizontal plane (left) and the vertical plane (right) for the 8Khz octave band using data from [30].

3.5.4 RIR Topology Problems

RIR Analysis

It has been noted that the RIR's produced will attempt to accurately resemble a human head. This involves placing the sound source below the receiver, to resemble the mouth (sound source) below the ears (receiver). When taking real RIR measurements it is often not possible to get the source and receiver close enough due to the physical dimensions of the equipment used and therefore result in an unnatural distance between them (see section Additional Functionality). However, with using software such as Odeon, where sound sources are calculated from a point source, it is possible to move the source and receiver much closer together. Initially, the sound source was placed 1m off the ground and the receiver was positioned 0.05m above the sound source. Three RIR's positioned along the centre of the room at varying distances from the left wall can be seen in figure 25. (These RIR's are taken from grid positions 76, 77 and 83 respectively which can be seen in figure 31 and is explained in section RIR Locations)

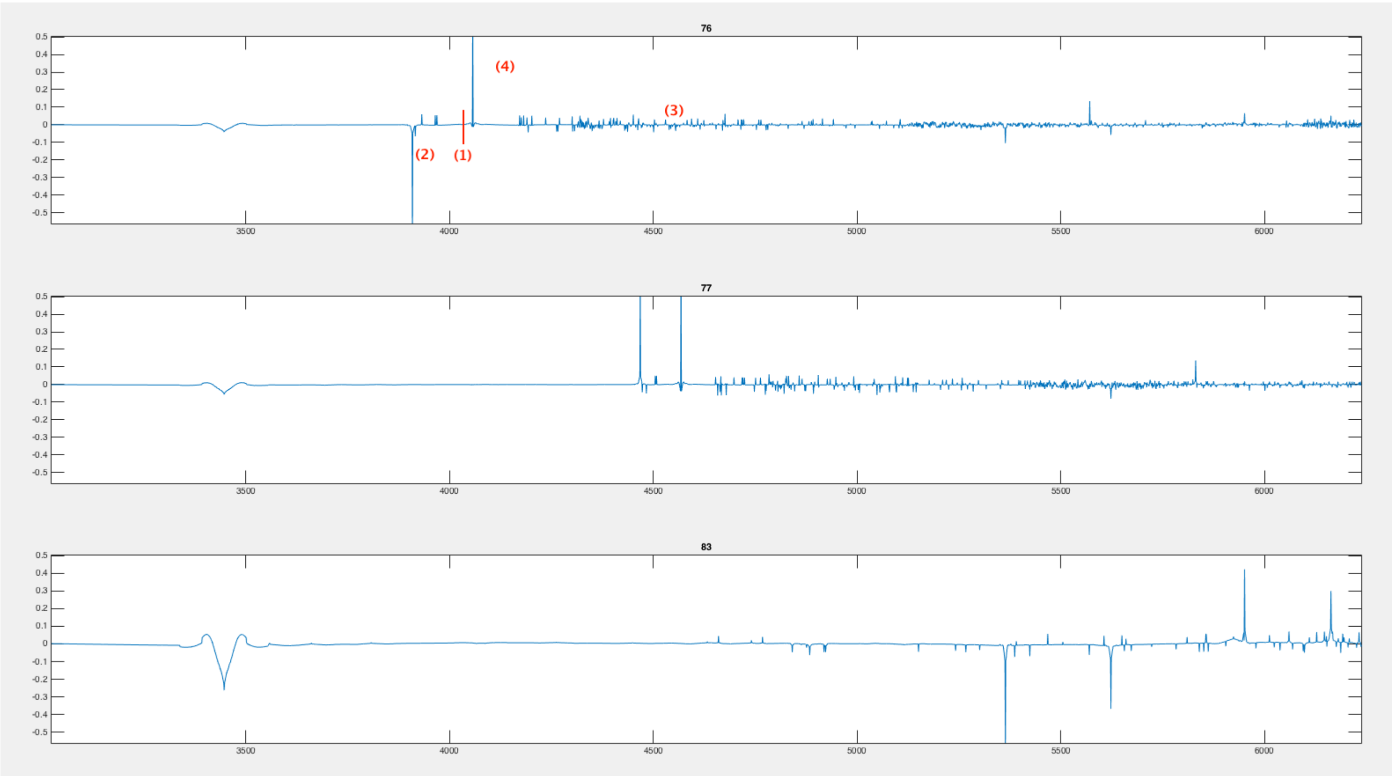


Figure 25: Initial RIR output with receiver 5cm above sound source showing RIR's from grid positions 76, 77 and 83 shown in figure 31

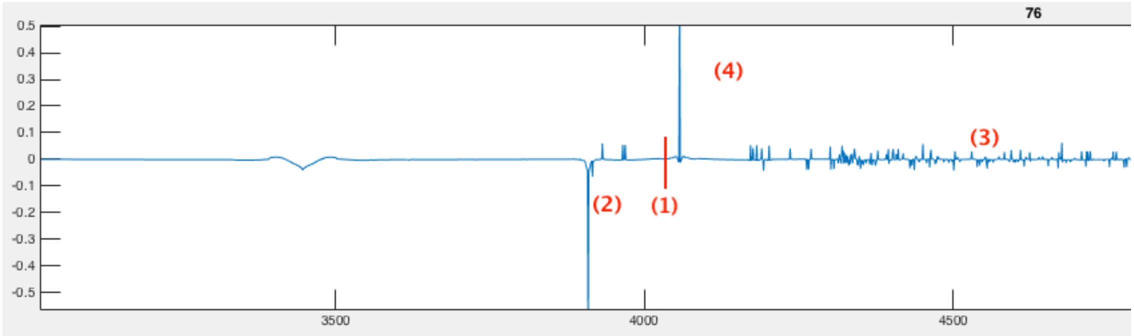


Figure 26: Initial RIR output with receiver 5cm above sound source from grid position 76

These RIR positions were chosen as the varying distance from the left wall can be used to estimate when to expect reflection other than that from the floor which will remain the same across each RIR. At the beginning of each plot, a dip of varying amplitude at the same instance in time can be seen. Without further inspection, this could be assumed as the direct sound from the source to receiver, which would be in the same place in time, as the source and receiver are kept the same distance across all RIRs, though the varying amplitude is questionable.

Figure 26 shows the first RIR from figure 25 which is used for the analysis of early reflections. The RIR's can be validated by calculating where expected reflections should be, using the following equation:

$$t_s = t_p - \frac{d}{c} \tag{1}$$

where:

t_s = Expected start of impulse (seconds)
t_p = Time of first recorded reflection (first peak) (seconds)
d = Distance between source and receiver (meters)
c = Speed of sound ($344m/s$)

Therefore, if this dip is assumed to be the direct sound which occurs at $t_p = 0.03448s$, where $d = 0.05m$, then the start of the impulse can be calculated to be:

$$0.03448s - \frac{0.05m}{344m/s} = 0.03433s = t_s \tag{2}$$

This is indistinguishable from the direct sound peak, which is understandable given the source and receiver are only separated by 0.05m. Now the expected start of this impulse is known, the expected first reflection due to the floor can be calculated:

$$\frac{2.05m}{344m/s} + t_s = 0.4029s \tag{3}$$

However, it can be seen that the first reflection occurs at (2), at a time of $0.039s$ suggesting that sound has travelled 1.6392m before reaching the receiver.

$$(0.039s - t_s) * c = 1.6382m \tag{4}$$

However, the shortest available distance a sound could travel other than directly from the source to the receiver is 2.05m (the first floor reflection).

As the source and receiver are placed 1.85m away from the left side wall (meaning sound has to travel slightly over twice this distance), a second reflection would be expected to occur at:

$$\frac{1.85m \times 2}{c} + t_s = 0.04509s \tag{5}$$

indicated by (3) in figure 26. However, it can be seen that no strong reflection occurs here.

After an email was sent to the technical department at Odeon [31], it was suggested that the small dip at the beginging of the RIR may be a bug in the version of Odeon being used. It could therefore be assumed that the direct sound is actually captured at the peak shown at (2) in figure 26 at a time of 0.03908. From this the new expected time for the start of the impulse could be calculated:

$$0.03908s - \frac{d}{c} = 0.03893s = t_s \tag{6}$$

It could then be assumed that the first reflection caused by the floor is the peak at (4) in figure 26 which occurs at a time of 0.04058s. However, this would suggest that sound has travelled 0.05532m:

$$(0.04058s - t_s) \times c = 0.05532m \tag{7}$$

meaning that it has reflected off a surface approximately 0.28m away, of which there is no surface.

It is also apparent that if the first peak other than the dip at the beginning of the file is taken to be the direct sound, it can be seen across the RIR plots shown in figure 25 that these would occur at different times depending on room position. This is also incorrect as it has been mentioned that the source and receiver distance is kept constant for all RIR simulations. The first strong reflection appears to be delayed in time the further away from the left side wall the RIRs are taken which would be accurate, however the timings are incorrect.

Experiment

After some further investigation, it was found that the Odeon Manual [12] states in the section **Minimum distance from receiver to closest surface**:

> *"If a receiver is placed very close to a surface then results will be sensitive to the actual position of the secondary sources generated by ODEON's late ray method. If such a secondary source happens to be very close to the receiver, e.g. 1 to 10 centimetres, this may produce a spurious spike on the decay curve, resulting in unreliable predictions of the reverberation time [...] to avoid this problem it is recommended that distances to surfaces are kept greater than say 0.3 and 0.5 meters"*

Though this is a problem said to be caused by secondary sources, it can be assumed that this issue could also be caused by primary sources. This was investigated by producing RIR's with the source in a stationary location with receivers varying in distance above the source. Figure 27 shows six RIR's produced using different distances between the source and receiver where the numbers 1 - 6 indicate the following:

Graph Number	Distance from floor	
	Source	Receiver
(1)	1m	1.05m
(2)	1m	1.15m
(3)	1m	1.30m
(4)	1m	1.50m
(5)	1m	1.60m
(6)	1m	1.70m

The following numbers (1) - (6) refer to the individual plots in figure 27.

A 5cm distance between the source and receiver shown in (1) was originally used. This plot can be seen to be greatly different from the rest (***Note:*** *The reason the RIR shown in (1) looks different to previously investigated RIR's such as those in figure 25 is due to the plot being zoomed in, causing it to look greater in amplitude. This was done so it could be compared to (2) - (6) whilst being on the same scale*). It is in this RIR that the 'dip' found in the previously investigated RIR's is contained, whereas the RIR's with a greater distance between source and receiver (2) - (6) do not.

Results

It is suggested in the Odeon manual (quoted above) that source and receiver be at least between 30cm and 50cm apart. Though a distance of 15cm (2) shows an RIR similar to the rest of the assumed correct ones, it is obvious that there is one strong direct sound but it is difficult to see the other expected direct reflections, whereas when the receiver is moved further away from the source to a distance suggested by the Odeon manual, the early reflections become more obvious, with clarity of strong peaks being more consistent through (4) - (6). The peaks in the RIR (5) (60cm) were investigated to assure their correctness, shown in figure 28. This RIR was chosen as the distance between the source and receiver for the real RIR's taken in Hendrix Hall was also

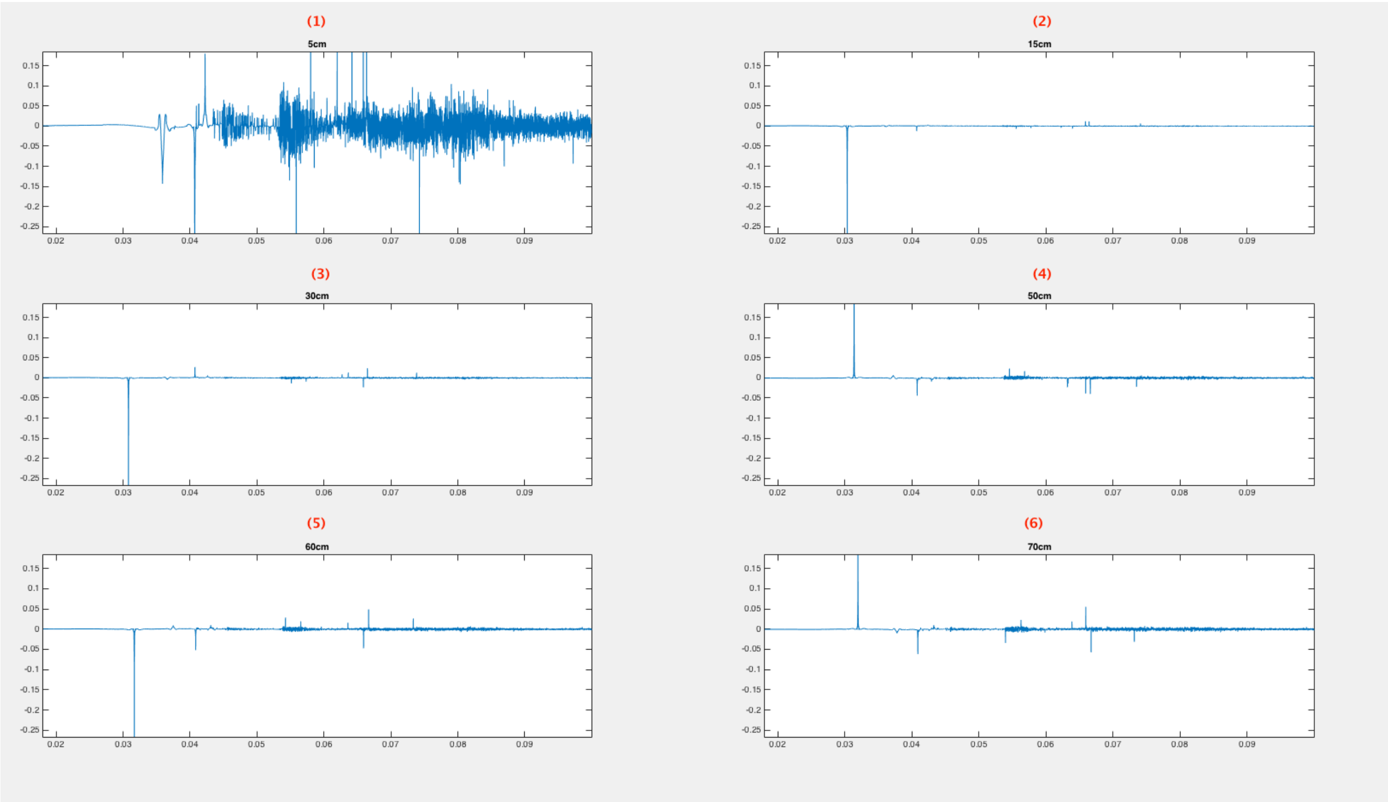


Figure 27: Plots of RIR's produced with the receiver placed at varying distances from the source, indicated by the title of each plot where **(1)**: 5cm **(2)**: 15cm **(3)**: 30cm **(4)**: 50cm **(5)**: 60cm **(6)**: 70cm

60cm. As the aim of achieving a more accurate human head topology was not possible due to this issue caused by Odeon, it was decided that for now, consistency is the next best thing.

The first peak (a) can be assumed to be the direct sound, present at 0.03168s, meaning the start of the impulse can be calculated using equation 1, where $d = 0.6$:

$$0.03168s - \frac{0.6m}{c} = 0.02994s = t_s \tag{8}$$

From this, it can be calculated that the peak at (b), which occurs at 0.0375s had been captured after sound has travelled 2.6m:

$$(0.0375s - t_s) \times c = 2.6m \tag{9}$$

This is the reflection from the floor which is 1.6m below the receiver, causing the sound to travel 1m to the floor and then a further 1.6m to reach the receiver. This is shown in **Path 1** in figure 29. The peak at (c), occurring at 0.04084s can be calculated as having travelled 3.74m which is the distance travelled by a reflection caused by the left side wall shown as **Path 2** in figure 29.

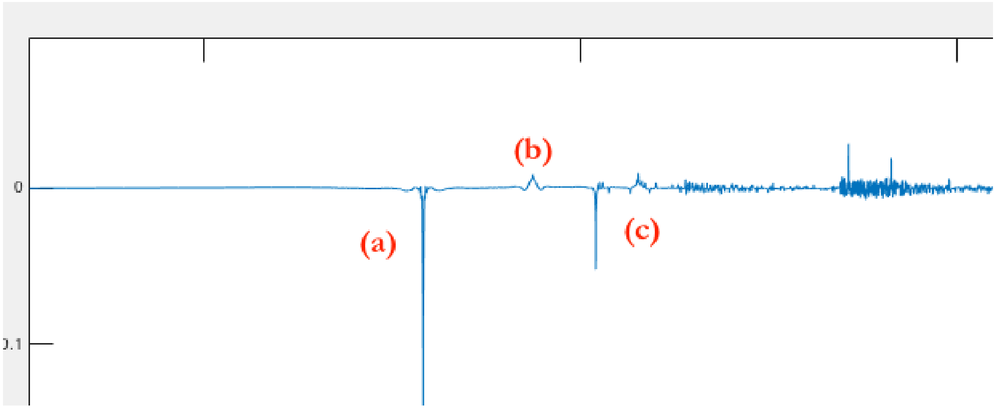


Figure 28: RIR produced when the receiver is placed 60cm above the source.

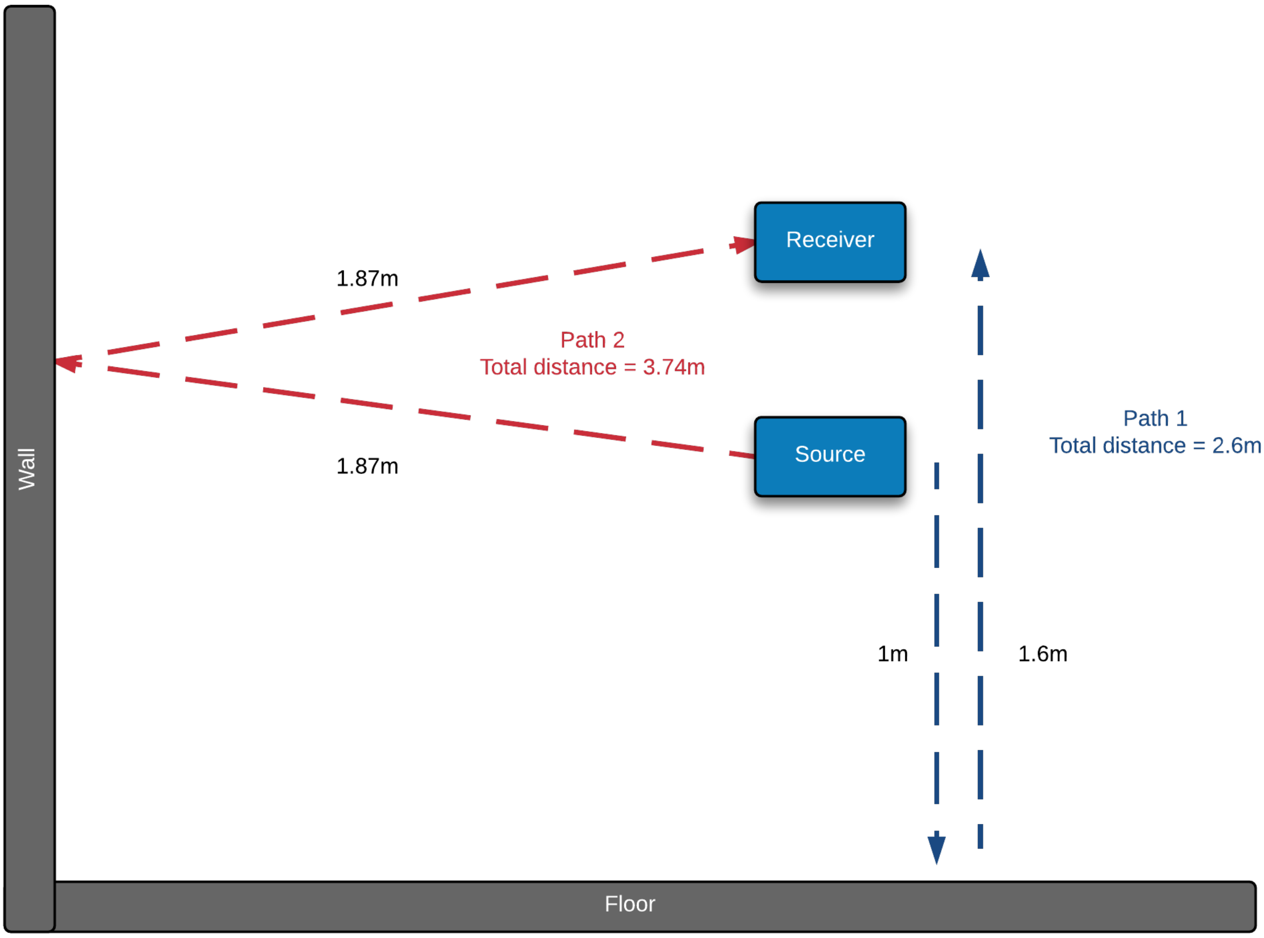


Figure 29: Illustration of the two early reflection paths due to the floor and left side wall when the receiver is placed 60cm above the source.

It can therefore be stated that the RIR's with a distance of 60cm between the source and receiver are accurate and can be used to produce a mass of RIR's.

3.5.5 RIR Locations

As discussed in section User Test Planning, RIR's were to be taken every 1m starting from the centre of the room and form a grid (rectangular or square). However, there were some positions within the room that would cause some RIR positions to be invalid. The location of these invalid positions could be found anywhere the sound source and receiver were placed closer than 0.75m to a surface. As the distance between the centre of the VSS (where the listener would be located) and the surrounding loudspeakers is approximately 1.5m, it would be impossible to reproduce a sound that has to travel less than this distance in the virtual space at the correct time, thus direct reflection from surfaces closer than 0.75m cannot be reproduced in time. When taking this into account, figure 30 shows the positions that would be invalid due to them being placed too close to a surface, marked as U, V, W and Z. W represents a large space of invalid RIR positions and they would all be too close to the table surfaces.

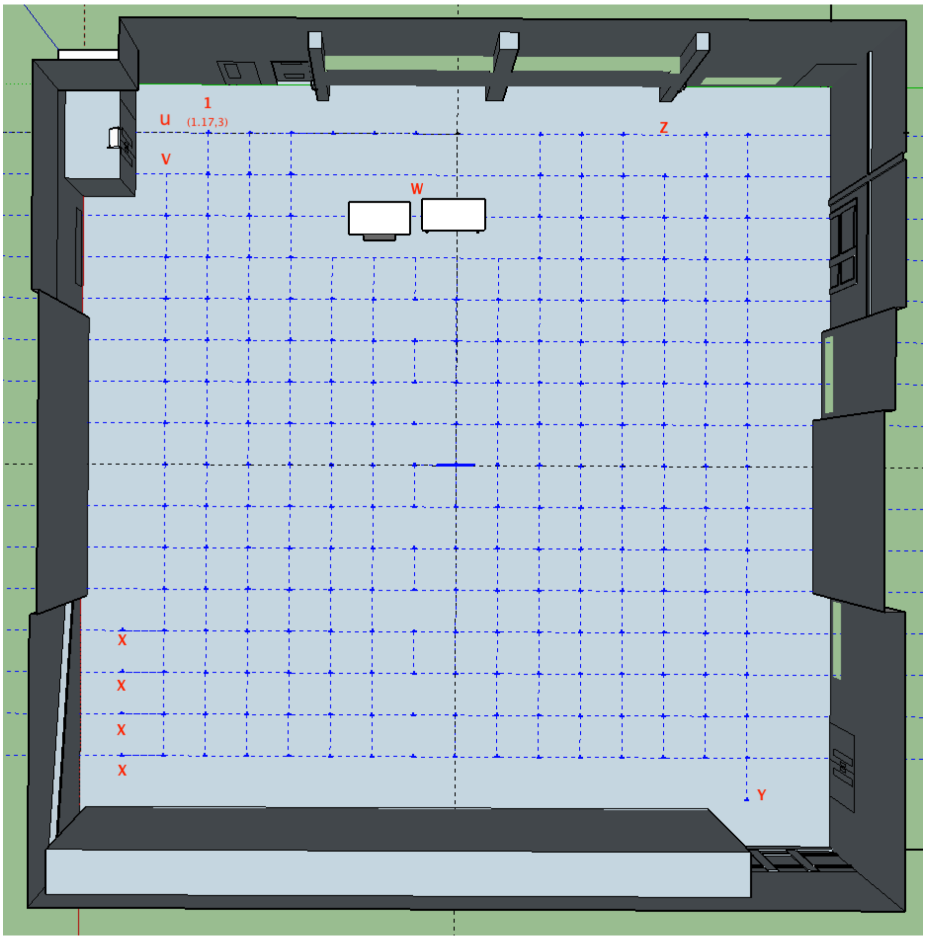


Figure 30: Top down view of the Hendrix Hall model indicating the RIR positions that are invalid due to being too close to a surface (U, V, W and Z), where other RIR's that are valid are not included as it would cause inconsistency (X and Y)

It was decided however that these RIR's would still be produced and used as part of the final system. If they were not included, there would be large gaps in the space where the user would have to jump between, thus be unable to produce what would feel like a consistent path.

It was also later discovered (see section **??**) that the latency due to the software used would mean that the beginning of the RIR's would have to be trimmed anyway, avoiding this issue altogether.

There are several RIR's in figure 30 labeled X and Y. These positions would still produce valid RIR's, however they were not included in the final grid as they would cause inconsistencies in the positions available to the user. All other blue crosses in figure 30 were also included in the final RIR grid, producing a grid 15x16, totalling 240 available positions. Figure 31 shows the final maximum available RIR's locations.

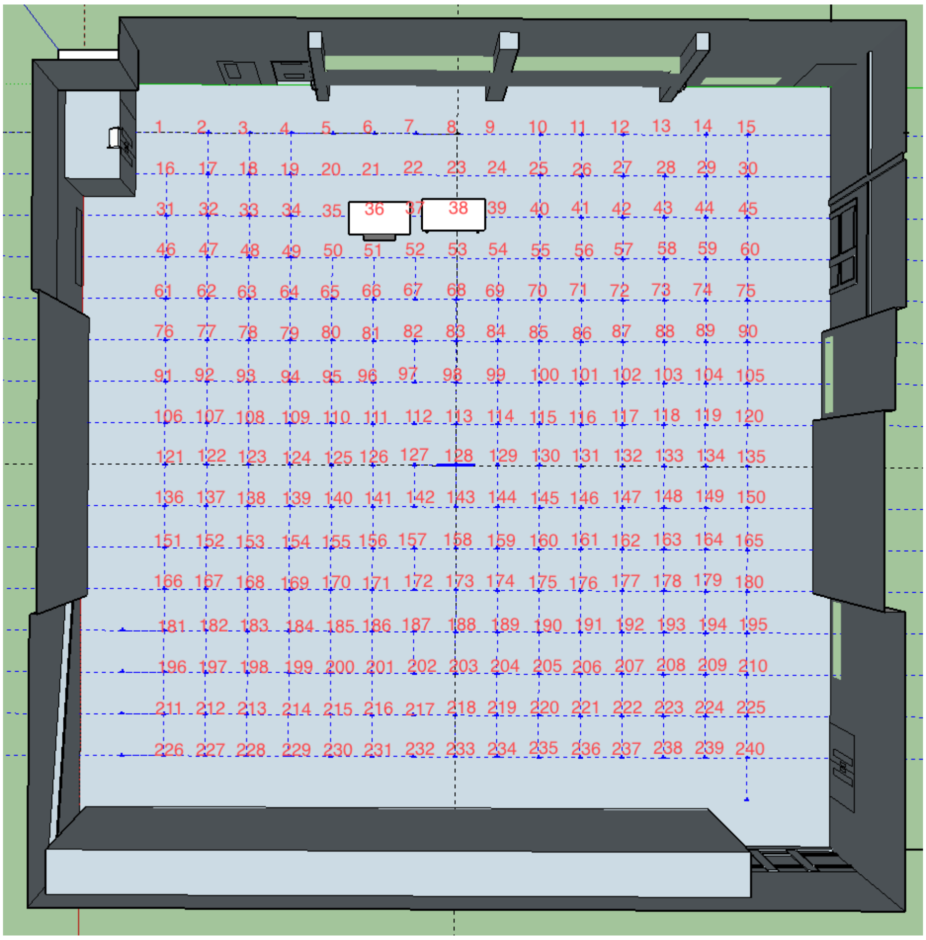


Figure 31: Top down view of the Hendrix Hall model showing the maximum number available RIR locations.

3.5.6 RIR Rendering

The source and receiver positions can be set using a .SouRecScript file, which contains information regarding the Cartesian coordinates of both sources and receivers within the room as well as the sound source directivity pattern and the direction in which the source is facing (available here or file 4.2.2). This made rendering 240 RIR's almost effortless. However, once the script had set the locations of the source and receivers, each source and receiver had to be paired up individually. This is because Odeon provides a **Job List**, where any number of RIR's can be rendered one after another automatically, allowing the process to run without user interaction. For this to work, the user must tell Odeon which receiver to use for each sound source. Figure 32 shows the job list menu where each source has to be connected to each receiver, using a series of drop down menus and scroll bars. This part of Odeon is clumsy and in no way designed to make the process fast and simple. This had to be done for all 240 RIR's and then again for the other three sets of RIR's used for facing different directions within the room, totalling 960 RIR's.

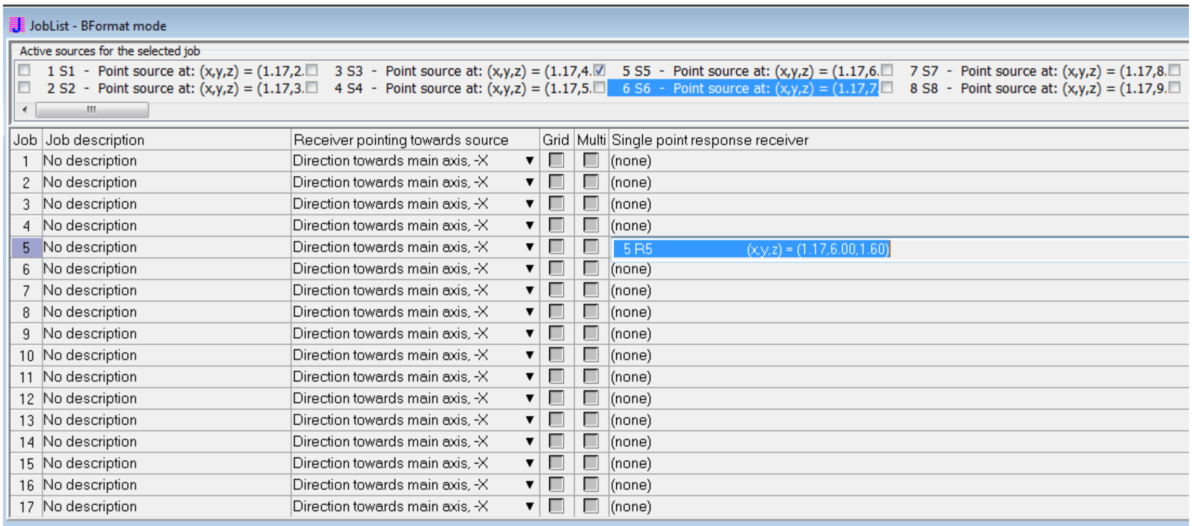


Figure 32: Source and receiver pairing window in Odeon

A naming system was used for the RIR files to indicate which grid positions they were taken from and which direction they were facing. For example: '008_0.wav' is an RIR from grid position 8 with 0° rotation (facing the front of the room) where '128_90' is taken from grid position 128 and has a 90° rotation (facing the left side wall).

3.6 - Producing Different RIR Grids

As described in section User Test Planning, user test #3 requires there to be a number of available RIR grids to choose from, each containing RIR locations separated by a fixed distance. It was decided that there would be a total of 5 grids to choose from, the first (the one already produced) containing RIR's separated by 1m, the second grid containing RIR's separated by 2m and so on. Four Matlab algorithms were written (fully commented code available: renameSortFiles.m) to extract the appropriate RIR's required for each new gird, and rename them starting from 001 up to the maximum number of RIR's in each grid. This made it easily integrable with the software described in section Software.

The four grids can be seen in figures 72, 73, 74 and 75 respectively in Appendix B, showing which RIR locations from the original grid they contain. The RIR files for all 5 grids can be found in file 3.2

3.7 - RIR Calibration

As both sets of RIR's that were to be used for user test #1 were produced using different methods, it was imperative that they were calibrated to prevent the difference in level from influencing the participants perception of movement. As the real RIR's were louder than the synthetic ones and the fact that there were far fewer real RIR measurements than synthetic ones, the real measurements were reduced in level to match those produced by Odeon.

3.7.1 Matlab

Method

In an attempt to perfectly reduce the level of the real RIR's to match those of the synthetic ones, using Matlab, a multiplier was calculated by taking the RMS level of both the real and synthetic RIR's (illustrated using the following pseudo code):

```
multiplier = RMS(real)/RMS(synthetic)
```

This value was used to multiply each of the samples in the real RIR to reduce each of the levels to that of the synthetic RIR. The full matlab script used can be calibrateALL.m can be viewed or found in file 2.

Figure 33 shows the plots of both of the original RIR's (top = real, centre = Odeon) and the calibrated version of the real RIR (bottom). As it can be seen, the peaks of the calibrated real RIR fall within the same range as the synthetic RIR (-0.02 to 0.02). However, when these RIR's were convolved with an audio sample taken from OpenAir [32] (available here or file 1.1 as `Singing.wav`), it still sounded significantly louder than that of the synthetic RIR convolved with the same audio sample.

This can be heard in audio examples: `calRMSConv.wav` (calibrated real RIR) and `odeonConv.wav` (Odeon RIR) found here or in file 1.2

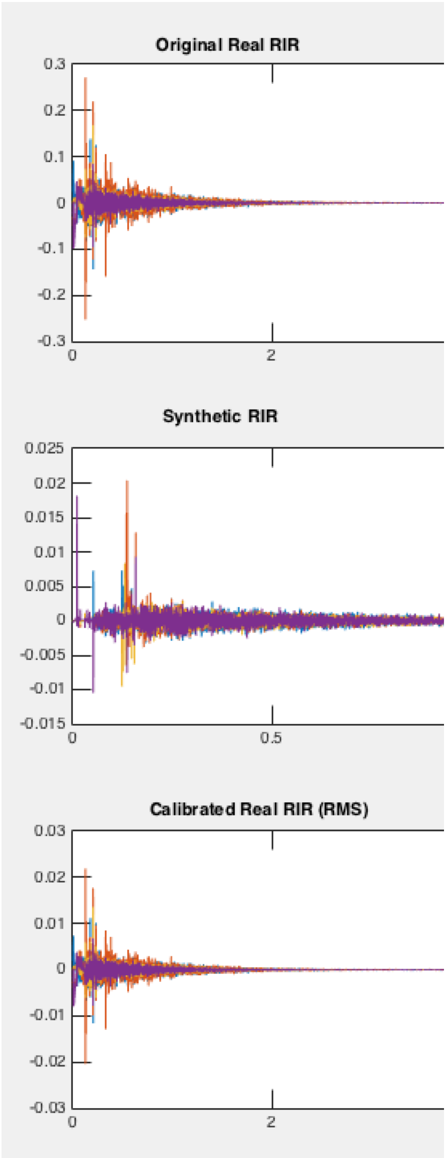


Figure 33: Plots of the original real RIR (top), the synthetic RIR (centre) and the calibrated real RIR (bottom) using the multiplier calculated using the RMS ratio value, showing that the peaks of the calibrated RIR sit approximately within the same range as the synthetic RIR (-0.02 to 0.02). (All 4 channels for each RIR are shown as they will not all sit within the same amplitude limits, hence the different coloured waves in each plot).

As can be seen in figure 34, the amplitude of the peaks in the signal convolved with the calibrated RIR are just over double those in the synthetic RIR convolved with the sample thus is perceived as louder.

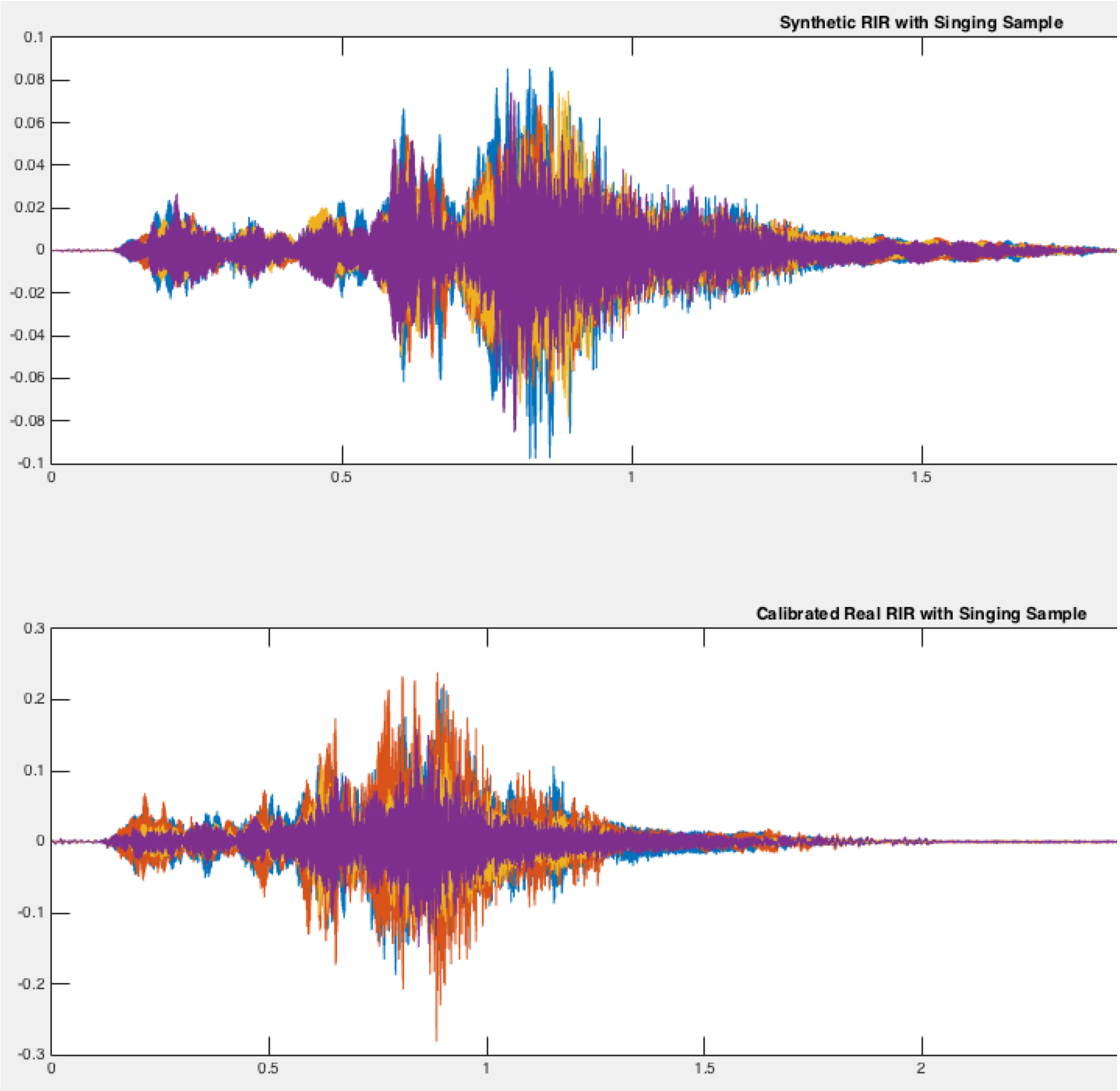


Figure 34: Plots of the synthetic RIR (top) and the RMS calibrated real RIR (bottom) convolved with an anechoic audio sample showing that the peaks of the calibrated RIR convolved sample sit approximately in the range double that of the synthetic RIR convolved sample, thus is perceived as louder.

Several manual values for the multiplier were tried instead, starting from 0.7 down to 0.2. It was found that a value of 0.3 normalized the real RIR's to the desired level. Figure 35 shows that the level of the manually calibrated RIR falls within a range of values half the magnitude of those found in the synthetic RIR, which then makes the magnitude of the manually calibrated audio sample (shown in the bottom of figure 36) approximately the same level as the audio file produced using the synthetic RIR.

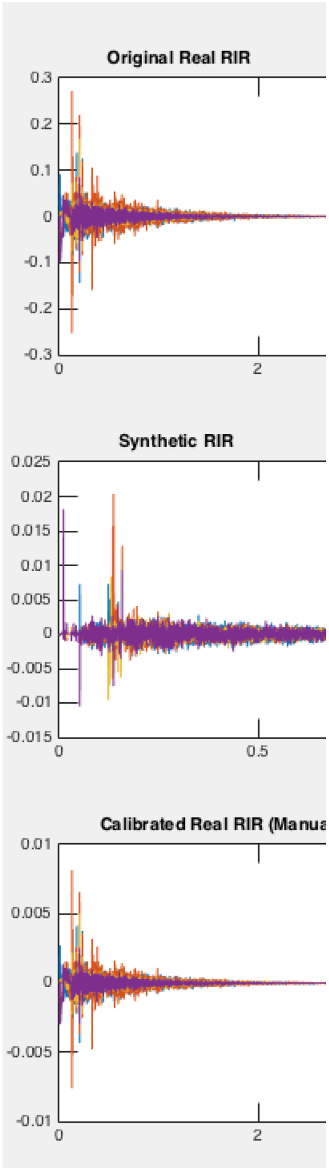


Figure 35: Plots of the original real RIR (top), the synthetic RIR (centre) and the *manually* calibrated real RIR (bottom) showing that the peaks of the manually calibrated RIR sit within half of the range of the synthetic RIR.

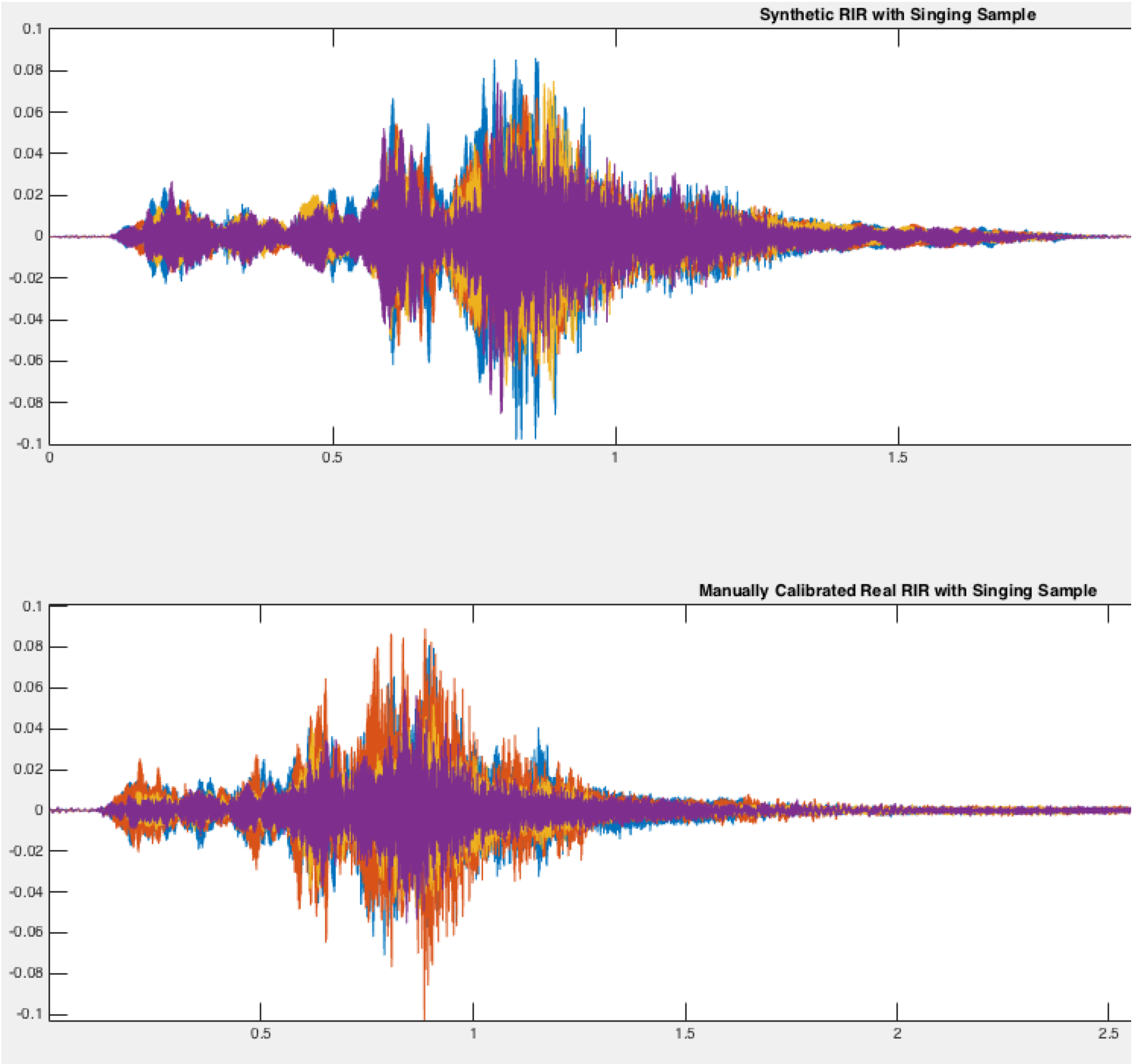


Figure 36: Plots of the synthetic RIR (top) and the RMS calibrated real RIR (bottom) convolved with an anechoic audio sample showing that the speaks of the calibrated RIR convolved sample sit approximately in the range double that of the synthetic RIR convolved sample.

Issue

After listening to the convolved audio file more closely, it could be heard that the tail of the reverb seemed to increase in volume, producing what sounded like a ‘swelling’ effect. Figure 37 shows the reverb tail of the manually calibrated convolved audio sample, where small pulses (occurring approximately every 0.05s), thought to be the culprit of this undesired trait, can be seen (mostly in the channel shown in yellow). Further investigation did not reveal the cause of the pulsing. The effect of the pulse can be heard in the audio sample pulseExample.wav found here or in file 1.2.

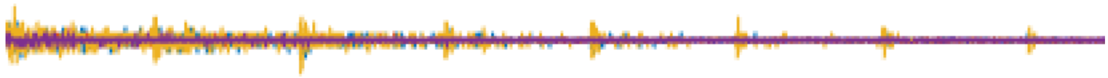


Figure 37: Reverb tail of the audio sample convolved with the manually calibrated RIR showing a pulse like pattern.

As the cause of the problem could not be uncovered within the time allocated, a less accurate but plausible manual calibration method was used.

3.7.2 Reaper

All of the real RIR files were imported into Reaper and send through a bus to a master channel in order to normalise all RIR's evenly. These were then compared by ear to a synthetic RIR and manually level matched. The closest match was by reducing the level of the real RIR's by 28.6dB. Though this method is less accurate, it prevented any unwanted errors in the signal such as those caused by using Matlab and allowed for the job to be done within a reasonable amount of time.

3.8 - Room Modelling and Synthetic RIR Rendering Summary

As stated in section Project Aims and Motivation, one benefit of using the direct RIR rendering method is that it is simple. The process of obtaining the desired RIR's has shown that though time consuming, no complex procedures were undertaken.

A benefit of using Odeon to obtain these RIR's is that it has allowed for the directivity of a human head has been modelled, thus providing a more accurate sound source directivity pattern than would have otherwise been achieved and was no more hassle to do than rendering RIR's using Odeons standard omni-directional point source.

However, other aspects of the Odeon software effected the accuracy of the RIR produced, such as the material list. Though the material list could be edited and built upon, the fact that its initial available options were limited instantly reduced the possibility for an accurate simulation of room acoustics. It has been shown that it is possible to achieve a close to desired result through the manipulation of the materials absorption coefficients, however this issue coupled with the already stated issues of geometrical acoustic modelling methods regarding their accuracy diminishes the plausibility of the system. In addition, the extra time spend on tweaking the resultant RIR's is not desirable, however this can be said for any modelling technique that involves simulation, thus is not restricted to this project alone.

Though it has been shown that producing 960 RIR's can be done sat in front of a computer, Odeon is not optimised for such a task. As stated in RIR Rendering, though it is possible to import a simple script to set all of the source and receiver locations, having to join them up one at a time takes the best part of an hour of repetitive point, clicking and dragging and for this project has to be done four times over. This undesirable software flaw would be simply fixed by adding the option to pair them up within the source and receiver script. In addition to this, though the required distance between the source and receiver in order to prevent erroneous calculations is larger than desired, it is still possible to place them closer together than it is when using physical equipment to measure real RIR's.

In summary, a large grid of 960 RIR's in 240 locations within a virtual space of choosing have been relatively easily produced, providing a good approximation of its room acoustics. The only real issue faced was software that is not optimised for such a task.

3.9 - SOFTWARE

As a software patch written in Max/MSP was already in use with the VSS, the idea was to extend this software patch to accommodate the newly proposed functionality. This section will first give a quick overview of the original max patch and then an overview of the newly produced software with a simple explanation of how it works, followed by a more detailed explanation of how the two main parts of the software were designed, highlighting the issues faced when doing so.

The software described was run on a 2012 Mac mini with 16GB of RAM, an Intel Core i7 processor running OSX 10.10.4.

All files javascript files referred to in this section along with the Max patch itself can be found in file 3 or here and video examples can be found here.

3.9.1 Software Overview

The Original Patch

The original max patch was used to convolve a real time audio signal with a set of four RIR's simultaneously, allowing the user to turn their head in the VAE through the use of an Oculus Rift as a head tracking device. Four positions within the VAE were available. To select one, the user (or an operator) selected an 'open' button which prompted a file navigation window. The four RIR files then had to be found (in the correct order of 0°, 90°, 180°and 270°) and opened one at a time, with a new file window opening after each file had been selected. A screen shot of this process is shown in figure 38. It was this part of the original patch that needed building upon as this process had to be automated.

Extended Patch Overview

Figure 39 shows an annotated top level view of the Max patch produced to take a user input, load the appropriate RIR files and convolve with a real time audio input. The annotated sections can be described as follows:

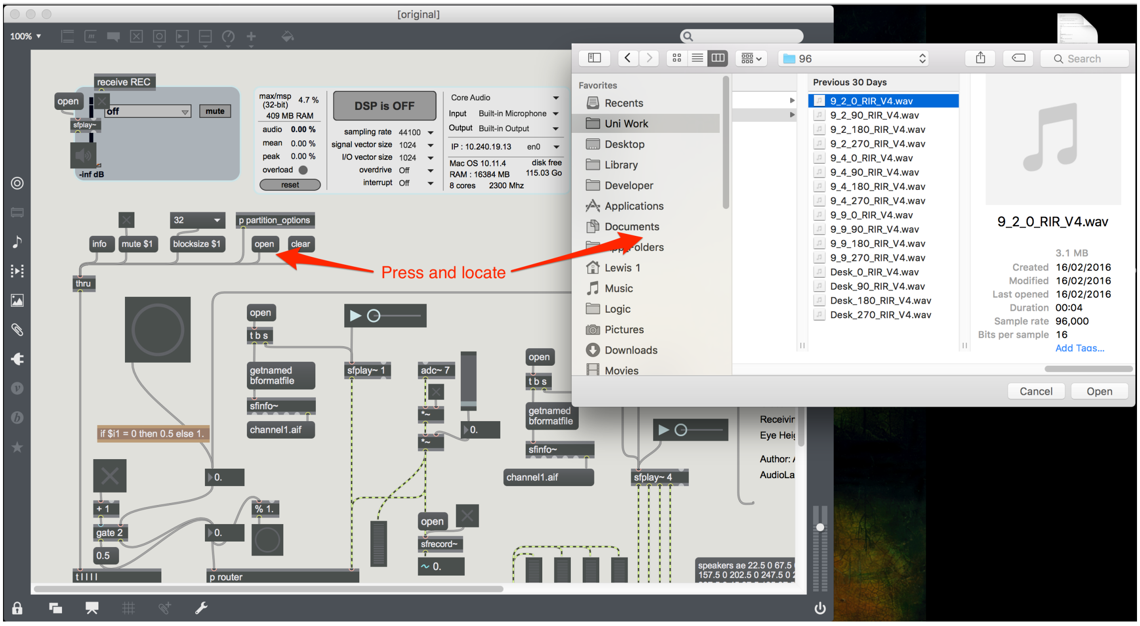


Figure 38: Original Max patch used in the VSS showing the file location window that pops up 4 times.

1: Two buttons used to reset the system and start the timer in section **3**. The settings patch extends to provide a range of options including which RIR grid to use.

2: Here, an audio file can be loaded into the system and used instead of a real time audio input. This is used in user test #3.

3: A timer system used to synchronise the output of data transfer (from **4.2** to **4.3**), panning between convolved audio and the movement of the user interface feedback system (see section Location Tracking).

4.1: Patches that send a user interface to an iPad which allows a user to select a location within the VAE. User interaction is monitored (in the form of screen coordinates) and sent to **4.2**.

4.2: Takes the coordinates of the user input and calculates which (if any) RIR file should be loaded into the system.

4.3: Three patches (extended versions of the Max patch explained in section The Original Patch) used to simultaneously convolve an audio signal (real time or audio file) with four directional RIR files. While one loads the next necessary RIR file the other two are used to simulate the movement of the user by panning the real time audio between the two currently running convolutions. This is how the user is moved along a path, explained in section Iteration 3 (Final).

5: An extended version of the real time head-tracking system used in the original patch, used to pan between the four directional RIR's to simulate head movement in the VAE.

The software implementation was split into two main sections: **Location Selection**, consisting of section **4.2** and **Mobility**, mainly consisting of section **3 and 4.3**, both of which are explained in

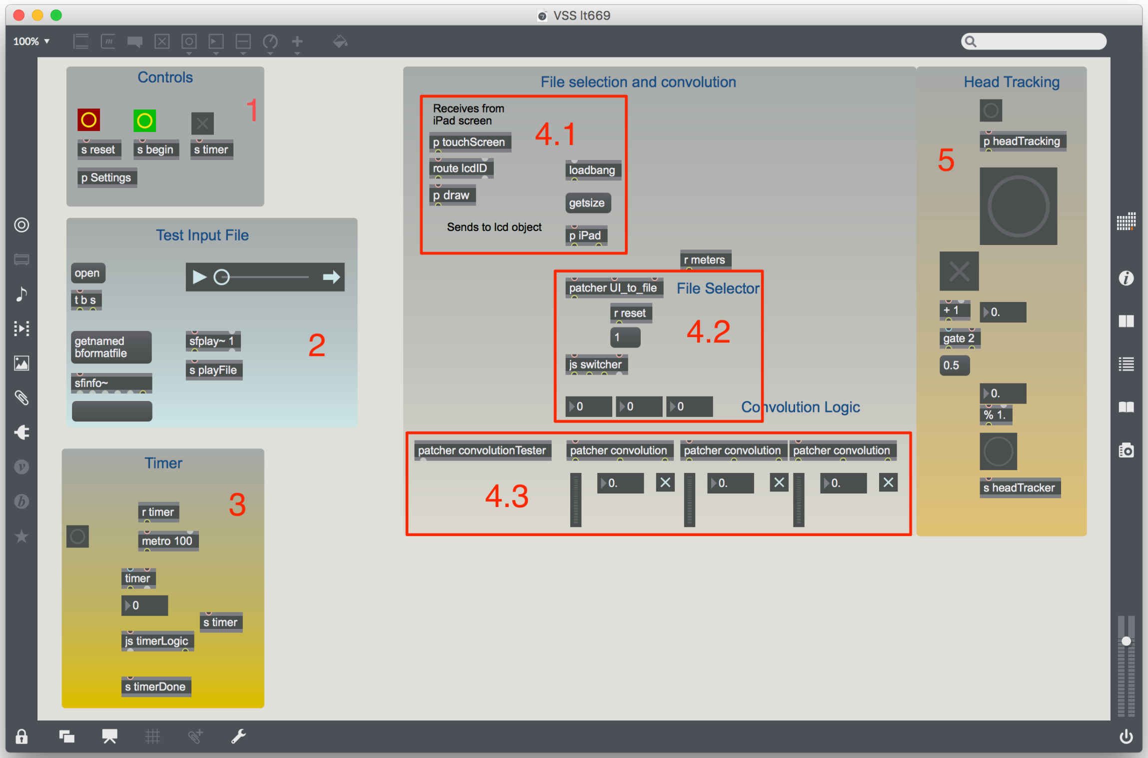


Figure 39: Top level view of the extended Max patch where: **1)** Control section, **2)** Audio file input, **3)** Timer function, **4)** User interface and Convolution section, **5)** head tracking input section

detail in the following sections.

3.9.2 Location Selection

In order to allow the user to move themselves around the room, they had to be presented with a means of doing so. This involved presenting the user with an interface (UI) that resembled the virtual space available to them on which they could select their location. The coordinates of the selected locations then had to be convert into a format which could be interpreted as a file name, indicating which RIR in the grid of available positions to use. This could then be passed to the rest of the system to load the appropriate files. A simple block diagram of the user interface part of the system is shown in figure 40.

In Max, the 'lcd' object is used for this function. This object presents a quadrilateral of variable length and height with the ability to output its dimensional information by sending a 'getSize' message to its input, as well as output the coordinates of a mouse click/drag. Figure 41 shows the lcd object with its inputs and outputs represented by section 1 in figure 40.

The outputs from the lcd object are sent to the patch 'UI_to_file' (located at the top of section **4.2** in figure 39) which contains a JavaScript file called 'loadFilesLogic'. This JavaScript file converts the

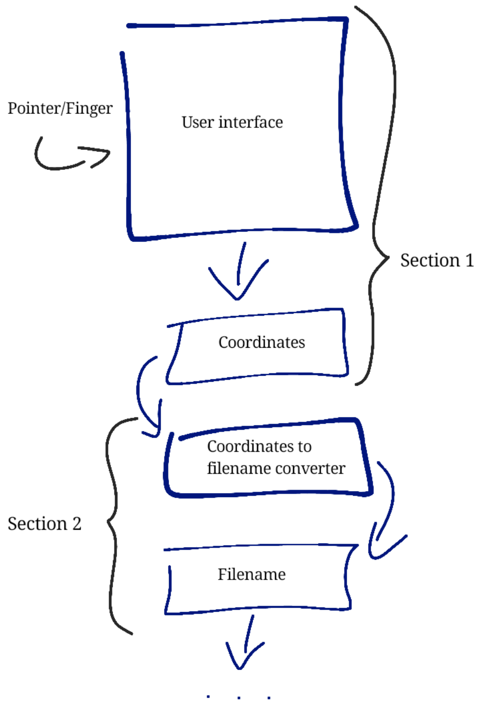


Figure 40: Flow diagram of the location selection software design. **Section 1** indicates the user interface section where coordinates are recorded and **Section 2** takes these coordinates and finds the appropriate RIR file.

(x,y) coordinates into an appropriate file name, by taking into account the size of the lcd screen and how many RIRs there are per meter.

This lcd screen could also be displayed on an iPad which was used for remote user interaction. Initially, the more popular and supported application Mira [33] was going to be used. This allows for a selected portion of a Max patch to be displayed on an iPad, allowing the user to interact with the selected part. However, the lcd object used to track coordinates was not supported by this application and thus did not appear on the iPad. Instead an alternative was found. Cycling74, the same company that produces Max/MSP and Mira also created an app for the iPhone and iPad called c74 [34] and allows for the creation of an independent interface that can control objects within the max patch. The operation of this section is explained further in section Location Tracking

Figure 42 shows a section of 'UI_to_file' that contains the 'loadFilesLogic' javascript. It can be seen that it takes 5 inputs: UI (x,y) coordinates , (x,y) lcd dimensions and a number representing how many rows and colums of RIR locations there should be available. This last value is calculated by sending a number from 1-5 (desired distance per RIR) to the fifth inlet of the UI_to_file patch (on the far right). As the maximum number of RIRs that would be available per length of the room is 15, 15 is divided by the input and rounded to the nearest integer, giving the number of rows and columns the lcd screen should be split into (the exact number of rows and columns is calculated later in the javascript file). This information can then be used to determine in which *'section'* (row and column) the user is currently located based on their coordinates, allowing it to load the appropriate RIRs that are available in that section.

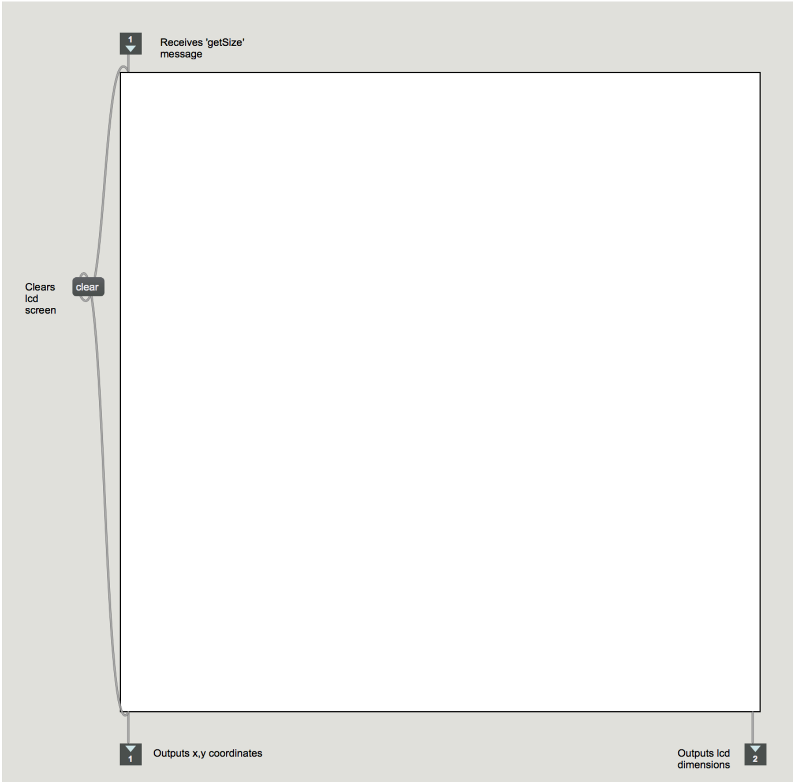


Figure 41: 'lcd' screen object in Max used for monitoring user interaction.

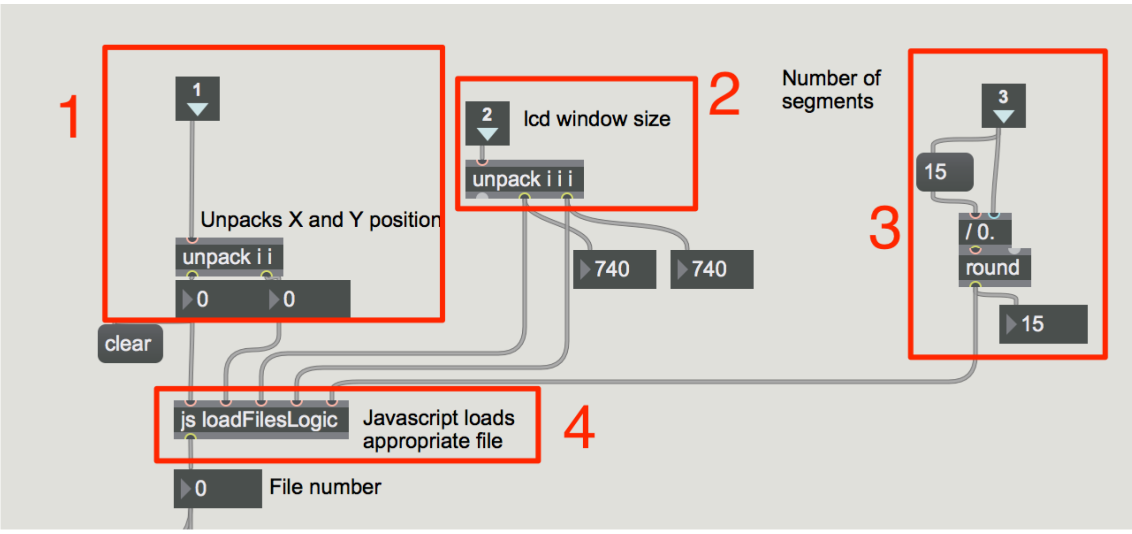


Figure 42: Screen shot from Max showing a section from 'UI_to_file' that converts the user interface coordinates into the appropriate file name. 1) Coordinates from lcd object 2) Dimensions of lcd object 3) Number of segments the lcd object should be split into due to the number of RIR's per meter 4) JavaScript file that produces an appropriate file name given the input data.

File name JavaScript: 'loadFilesLogic.js'

The first section in the JavaScript simply stores the data from different inputs to different variables that are used throughout the rest of the code as follows:

`xPos`:	user x coordinate
`yPos`:	user y coordinate
`windowSize[0]`:	lcd length
`windowSize[1]`:	lcd height
`numberOfMeters`:	Integer to calculate rows/columns

As can be seen in figures 72, 73, 74 and 75 in Appendix B, there are not always the same number of rows as there are columns in each of the grids, meaning that the location at which the user is placed has to be calculated slightly differently for each one. The `if else` statements on lines 1 to 14 shown in figure 43, are used to calculate the users position by taking the selected x and y coordinates from the lcd screen and dividing them both by the x and y dimensions of the lcd screen respectively, giving a percentage of how far along the x axis and y axis of the room they are. By converting their position to a percentage instead of just using coordinates allows for the lcd screen to be resized whilst keeping the correct ratio of the RIR grid meaning it can be used on different sized screens in the future (see section Further Work). These percentages are then multiplied by either the number of rows (x axis) or columns (y axis) that are required for the desired RIR grid, giving an approximation as to which section of the RIR grid they are in. The number of rows and columns that are required are calculated by using the value stored in `numberOfRows` and adding or subtracting 1, depending on the shape of the grid being used. The following table indicates how many rows and columns there are given the inputs to the javascript:

Distance between RIR's (m)	Calculation	numberOfMeters	Rows	Columns
1	15/1	15	15	16
2	15/2	8	7	8
3	15/3	5	5	5
4	15/4	4	3	4
5	15/5	3	3	3

The RIR position that the user is closest to is then calculated by rounding the approximated section position calculated in the `if else` statements, to the nearest integer, shown in lines 17 and 18 in figure 43. Lines 21 to 26, create an initial offset to prevent the lcd screen from starting at location (0,0), as this is not a physical point in the room itself, thus preventing errors from occurring when searching for the appropriate RIR file.

```
//Split into sections
if(numberOfMeters == 3 || numberOfMeters == 5){
 //Even grid for 3m and 5m
 xPosition = (xPos/windowSize[0])*(numberOfMeters);
 yPosition = (yPos/windowSize[1])*(numberOfMeters);
} else if (numberOfMeters == 4 || numberOfMeters == 8){
 //4m separation requires different x,y coordinate scaling
 xPosition = (xPos/windowSize[0])*(numberOfMeters-1);
 yPosition = (yPos/windowSize[1])*(numberOfMeters);
} else{
 //Extra row for others
 xPosition = (xPos/windowSize[0])*(numberOfMeters);
 yPosition = (yPos/windowSize[1])*(numberOfMeters+1);
}

//Round to nearest value
xSection = Math.round(xPosition);
ySection = Math.round(yPosition);

//Start the lcd grid sections from column 1 row 1 instead of column 0 row 0
if(xSection == 0){
 xSection = 1;
}
if(ySection == 0){
 ySection = 1;
}

```

Figure 43: Code used to calculate user location

In figure 44, lines 6 to 26 contain one large if statement. This essentially prevents the program from calculating, searching for and trying to load a file if the same one is still in use, thus saving computation time. This is done by comparing the section of the grid the user is currently in (`xArray[0]/yArray[0]`) with the section that was stored the previous time their location was calculated (`xArray[1]/yArray[1]`). If the location of the user has changed, two more conditions are checked in lines 8 and 14. This ensures that only the section that has changed is updated, ie, if the user has moved to the left, only the X axis section is updated.

Finally, the appropriate file name is calculated. Two different algorithms are used depending on which RIR grid is being used. Both essentially multiply the number of positions there are per row by the number of the row the user is located. This is then added to the number of the column that the user is located, thus giving a number between 1 and the maximum number of locations there are available. Both algorithms can be seen in lines 21 and 23 in figure 44.

```
//Store current location
xArray[0] = xSection;
yArray[0] = ySection;

//If either coordinate is changed search for new files
if(xArray[0] != xArray[1] || yArray[0] != yArray[1]){

 if(xArray[0] != xArray[1]){
  //Store previous value
  xArray[1] = xArray[0];
  X = xArray[0];
 }

 if(yArray[0] != yArray[1]){
  yArray[1] = yArray[0];
  Y = yArray[0]
 }

 //Output user location within grid
 if(numberOfMeters == 4 || numberOfMeters == 8){
  fileNumber = X + ((numberOfMeters-1)*(Y-1)); //Requires different algorithm for 2
m and 4m due to different grid shapes
 } else {
  fileNumber = X + ((numberOfMeters)*(Y-1));
 }
 outlet(1,fileNumber);
}

```

Figure 44: Code used to search for appropriate file name

3.9.3 Mobility Implementation

The following section explains how the system that was intended to be used to move a user around the virtual environment was designed but not be used in the final system. The reasons for this and the compromise made between system speed and desired functionality is discussed. In total there were 3 iterations of the system with 2 major design changes, all of which are extensions of the original Max patch used in the VSS, using Spat to load RIR files into the system and for convolution with a real-time audio input.

Iteration 1

Initially, the idea was to pre-load a grid of the closest RIR locations that surrounded the user and simultaneously convolve the real-time audio with all of the RIR files. This is illustrated on the left in figure 45 which shows annotated grid positions on the lcd screen labeled 1 - 9. The patch on the right shows the corresponding volume bars (also labelled 1 - 9) used to automatically vary the output level of each of the convolved signals, thus linearly interpolation the user between available positions in order to approximate the room acoustics in these unmeasured locations. This required a panning algorithm for each of the positions in the defined grid, an overview of which is shown in figure 76 in Appendix C, based on where the user has moved relative to the centre position essentially allowing the user to move freely.

When the user reaches one of the other locations in the grid, that location becomes the centre of a new grid, and the appropriate files are loaded around that centre positions through the use of a javascript file: gridLoading.js. This requires the system to simultaneously load 4 RIR files per location, of which there are 9, meaning 36 files are loaded into the system at once. Due to the time taken to load all of the files, the system ran too slow and could not be used for real time movement.

A video demonstrating the functionality of the proposed system can be found here.

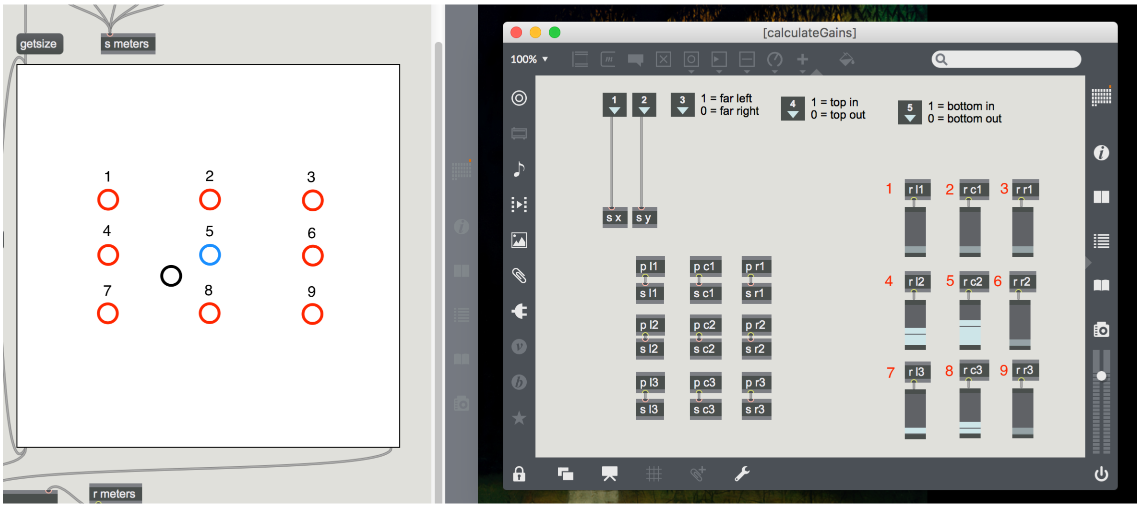


Figure 45: Illustration of how the interpolation system worked in iteration 1 with the computer version of the user interface on the left and the corresponding automated panning controls on the right.

Iteration 2

In an attempt to maintain the current implementation and solve the file loading time issue, a patch was built to pre-load all the RIR files. This involved running a convolution patch for each files loaded into the system. As the files would no longer need to be loaded into the system, the user interface section described in section Location Selection is used to decide which of the convolution patched to bypass and mute, and which ones to run. This means that only the grid of RIR's being used would be convolved with the audio signal, saving computational resources. Figure 46 shows part of the patch used to pre-load all the RIR files. The convolution patches were chained together in groups of ten. This was to allow the system to attempt to load the files

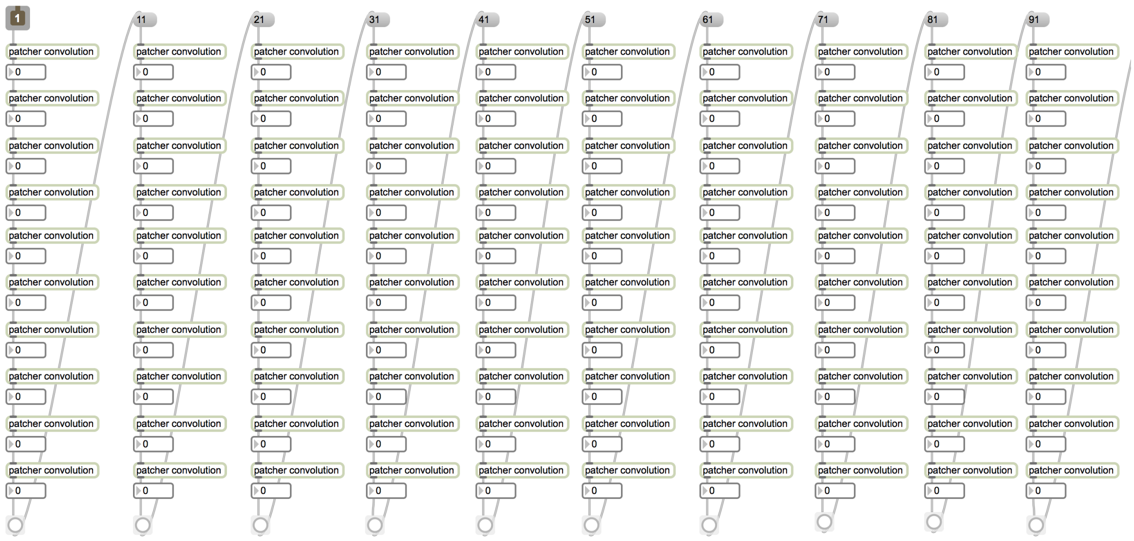


Figure 46: Chained convolution patches used to pre-load each RIR file. *Image taken from Max 6 as the version of this patch would not run in Max 7 at the time of writing.*

in chucks of 40 at a time, as opposed to the potential 960 files that would need to be available. Though it was possible to load smaller grids, such as a 9x9 grid used when locations are separated by 5m, the larger grids caused the system to run too slowly due to the amount of RAM that was required.

Iteration 3 (Final)

The previous iterations attempted to allow the user to move anywhere within the VAE in real time as they moved their finger around the screen, however due to time it took to load the RIR files into the system, this was not possible. It was therefore decided to find a compromise, where the user could draw themselves a path around the space and the system would move them along the path one RIR location at a time at a rate determined by a timer, allowing the system the load all appropriate files with ample time to prevent the system from intermittently freezing up.

The top half of figure 47 has been seen already in figure 42, however the bottom shows how the resulting file number is used (highlighted in the blue rectangle). It gets sent to a javascript file (storeLocations.js) that adds the file number to the end of a mutable array (an array that can be extended or truncated once created). As the file names are being stored, the corresponding files can be loaded when the system is ready, instead of trying to load them instantly as in Iteration 1. Once the javascript file receivers a message from the timer (highlighted as number 3 in figure 39), the number at the start of the array is sent to the output. The array is then truncated by 1, moving the next file number to the front of the array waiting for the next timer message. If a reset message is sent to the third inlet of this javascript (by pressing the first button in section 1 shown in figure 39), the arrays are cleared, enabling the user to start drawing another path if the previous path had not been completely travelled.

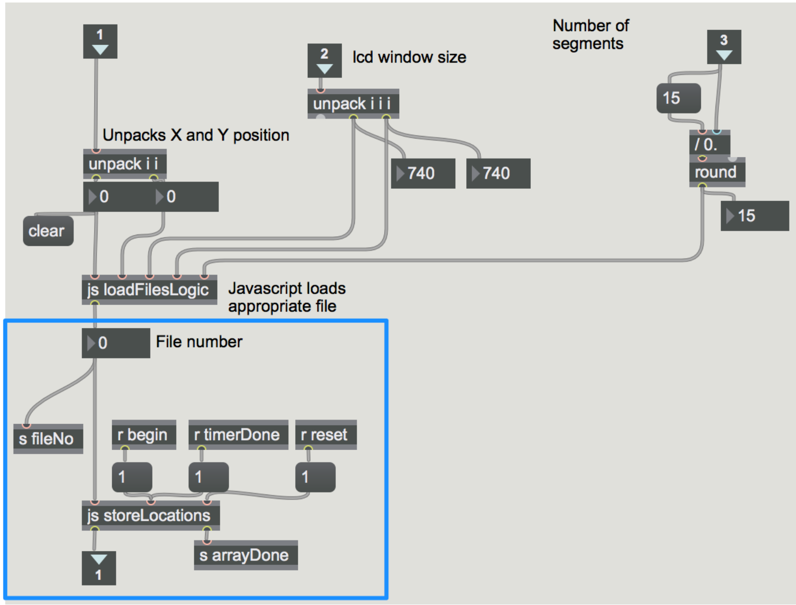


Figure 47: Highlighting the section of 'UI_to_file' that stores the file numbers and outputs them when a timer is done.

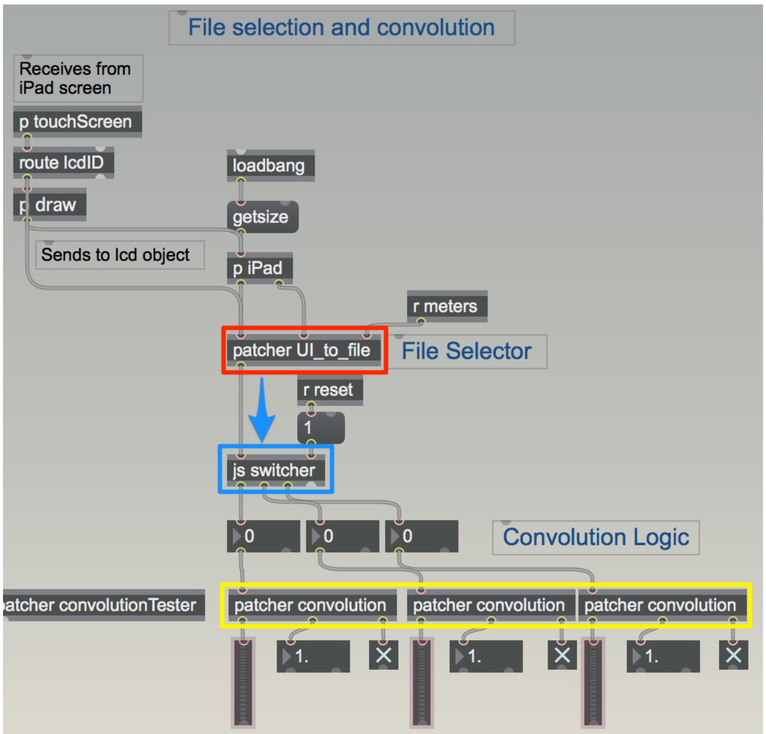


Figure 48: **Red:** Outputs file stored file number. **Blue:** Distributed file number to a different convolution patch each time. **Yellow:** Receives file number and loads appropriate file into the system.

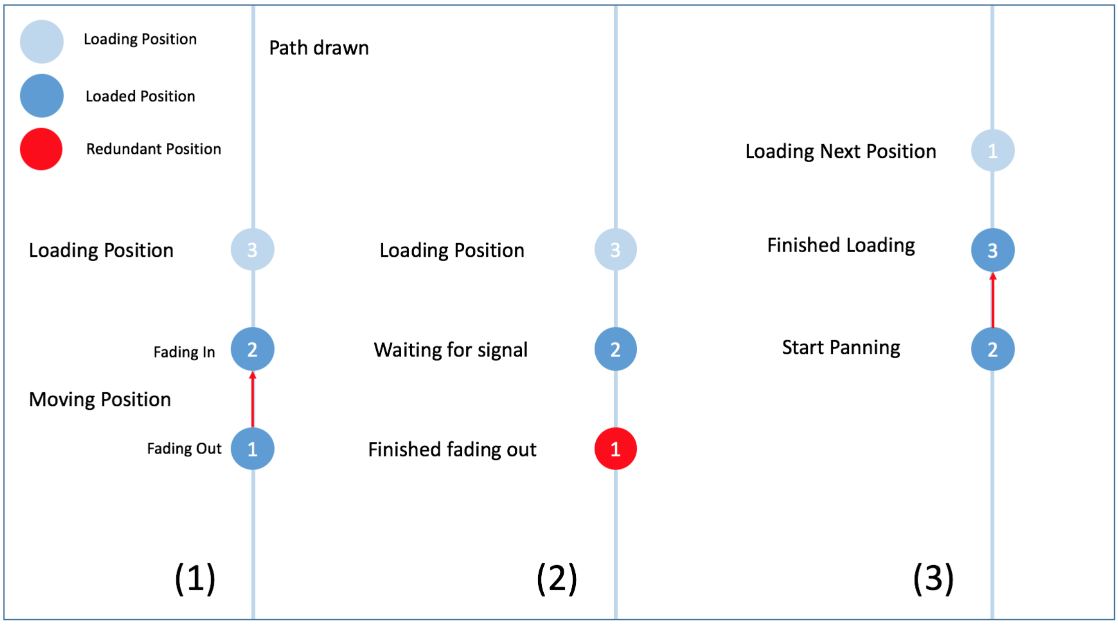


Figure 49: Illustration of how the three convolution patches work together to simulate movement, where the circles represent locations within the VAE and are numbered to indicate which convolution patch they are loaded in. **(1)** User is moved between two loaded positions while the third patch loads a new RIR file. **(2)** The user is held in the new position until the next position has been loaded into the system. **(3)** User is moved between the two available positions while the next position is loaded into the system.

Figure 48 shows the output from the 'UI_to_file' patch in the red rectangle (the patch that contains the contents of figure 47) running into a new javascript called 'js switcher' in the blue rectangle. This simple script ensures that each new file number is sent to a different convolution patch (shown in the yellow rectangle). This means that while a new file is being loaded into one of the convolution patches, the other two can be used to amplitude pan between two pre-loaded positions. Figure 49 illustrates this process.

The convolution patches have three outputs connecting to:
1: A level meter used to see which patch is being used.
2: A number box to check the level of each patch.
3: A toggle box used to see which spat convolution algorithms have been bypassed (a function that was later removed, explained in section Software Issues). These were used to monitor the functionality of the patch while testing.

To achieve the path following functionality illustrated in figure 49, each convolution patch contains its own level control algorithm. Figure 50 shows an overview of the first convolution patch which is a modified version of the original patch used in the VSS. The input to the convolution patch (orange) feeds the file number directly into a loadFiles.js javascript object. This simply prepends the number with the appropriate number of 0's (eg 1 becomes 001 and 28 becomes 028), then this number is concatenated with a file path pointing to where the audio files are located.

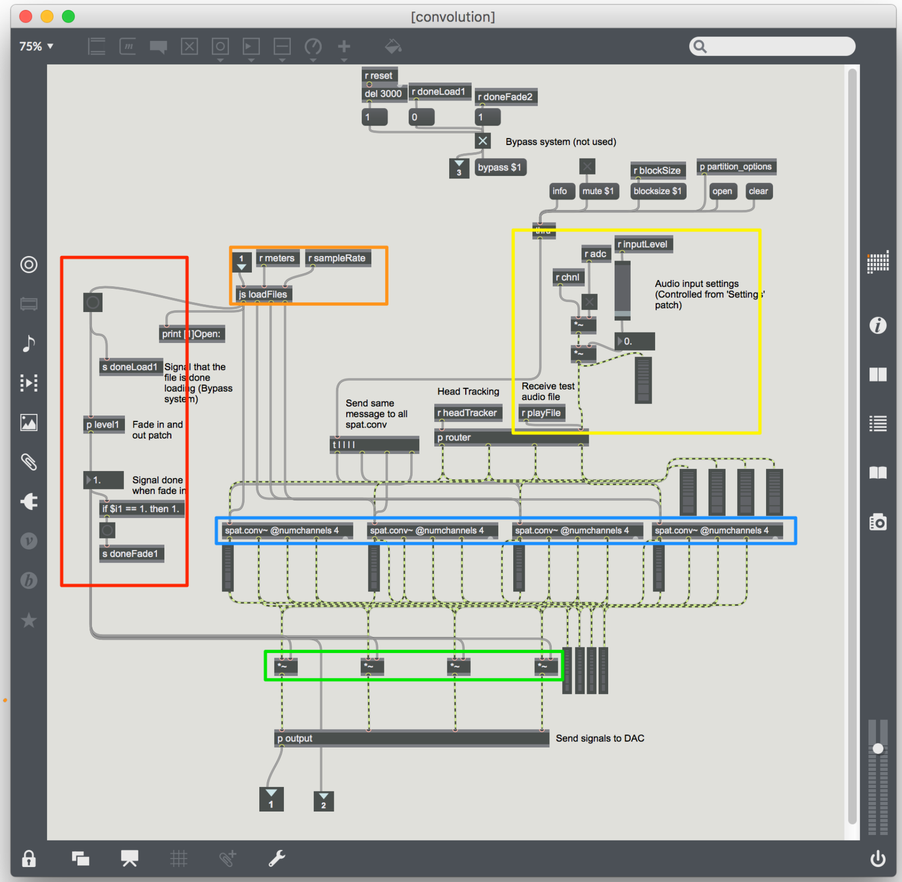


Figure 50: Overview of the one of the convolution patches showning: **Orange:** Input to the patch that receives the required file number, **Red:** Automated panning control, **Yellow:** Audio input section, **Blue:** Spat convolution objects, **Green:** Multiplication stage used for controlling the level of the convolved output.

This outputs four 'open' messages (one for each directional RIR file) followed by the appropriate file path to the spat.conv objects (blue). These are the objects that actually search for the file and load it into the system. By sending the prepending 'open' message, manually searching for a file is not required, thus automating the process. These objects convolve the loaded RIR file with whatever audio input is given at its inlet, in this case the real time audio input or audio file (yellow). These outputs are then sent through multipliers (green), used to fade the signal in and out with an automated volume control (red).

The automated volume controls receives a *'bang'* (signalling something has happened) when the 'open' message is sent form the 'loadFiles.js' object (orange). This bang is also sent to the convolution patch that is currently at full volume. This prompts the volume of the current patch to increase while the volume of the previous patch decreases, thus panning between the two signals.

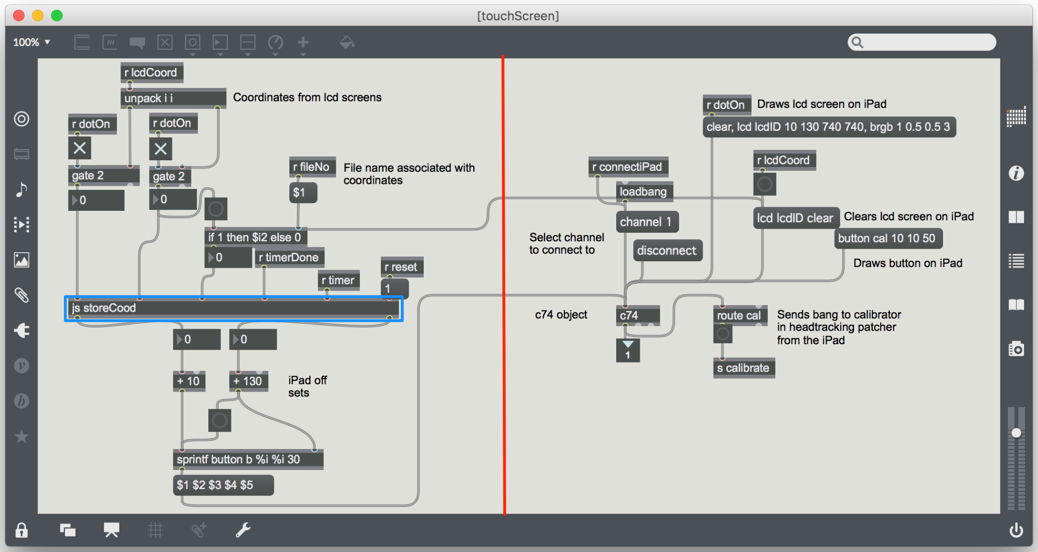


Figure 51: Screen shot of 'touchScreen', the patch used to produce the user interface on the iPad (right side) and to draw a dot on the iPad showing the user where they are located in the virtual space (left side). 'storeCood' javascript object highlighted in blue.

3.9.4 Location Tracking

Figure 51 shows the patch 'touchScreen' which contains two main elements. The right side is used to draw the user interface itself and the left side is used to draw a dot on the user interface to inform the user where they are positioned in the virtual space. Both are described in the following sections.

User Interface

As previously mentioned, the c74 application was used to display a user interface on an iPad, allowing the user to select their location within the space. This is created on the right side of figure 51 and is done by sending the 'c74' object a message that contains the type of object that should be presented, its coordinates in the space as well as size and colour, the results of which are shown in figure 52. The output of the 'c74' object returns information regarding the objects created in the user interface. In this case, the positions touched on the lcd screen will be sent from the iPad back to this 'c74' object. This information is sent to the input of the 'UI_to_file' patch. This allows the iPad screen to be used the same way as the lcd screen in Max is used (show previously in figure 41).

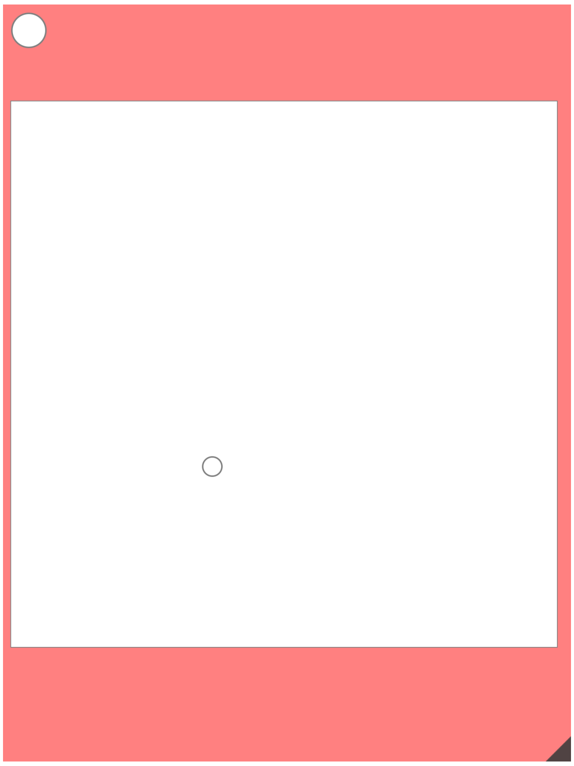
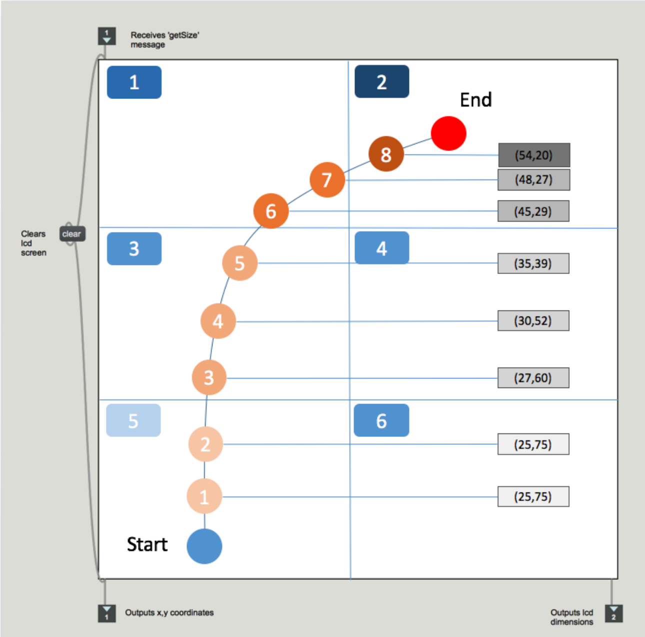


Figure 52: Screen shot of the user interface produced on the iPad. The large white rectangle represents the virtual space in which the user can place themselves, where the top of the rectangle is the front of the room (where the blackboards are in Hendrix Hall). The dot in the white rectangle is used to show the user where they are located in the space and the button in the top left corner is used for calibrating the head-tracking device.

Position Feedback

The left side of figure 51 contains a javascript file called storeCood.js (highlighted in blue) with a number of inputs. As mentioned above, the 'c74' objects can retrieve data from the lcd screen on the iPad. When the user selects a location, the (x,y) coordinates are sent to the first two inputs of the javascript file respectively. These coordinates are stored in a list along with the corresponding section that they are located in (third input), calculated using the method described in section Location Selection. Figure 53 shows a simple example, where the coordinates of a path spanning across 4 sections are stored in groups.

Input 4 and 5 receive two different signals routed from the main timer (labelled 3 in figure 39). Input 4 receives a signal every time the timer is done (ever 2.5 seconds) and input 5 receives a message every 100ms (the rate at which the timer increments). These are used to call the two main functions that are used to output the coordinates at the correct times, `findLength()` and `outputCoords()`. Each time the timer finishes (after 2.5 seconds), `findLength()` starts from the beginning of the list of stored coordinates and scans through them, calculating how many coordinates are located in the same section. For the example in figure 53, this would find a length



Point	Section	(x,y)
1	5	(25,80)
2	5	(25,75)
3	3	(27,60)
4	3	(30,52)
5	3	(35,39)
6	1	(45,29)
7	1	(48,27)
8	2	(54,20)

Figure 53: Simple example of the coordinates along a path being stored in groups depending on which section of the lcd screen they are located. *(The smallest number of sections possible is actually 9, but 6 are used here for simplicity)*

of 2 for positions 1 and 2 located in section 5. As the user is moved to a new RIR location every 2.5 seconds, `outputCoords()` will output the coordinates that were travelled in that section evenly within that amount of time. This is done by dividing 2500(ms) by the number of points within that section. For the example in figure 53, this would be $2500/2 = 1250$. Therefore, every 100ms the function checks to see whether it is time to output the stored coordinates, meaning the coordinates of position 1 and 2 will be output at times 1250ms and 2500ms. If we were to continue with the example, the next three positions, 3, 4 and 5 would be output at times 833ms, 1666ms and 2500ms, thus evenly spreading out the movement of the dot on the user interface screen meaning the dot should travel around the path at a continuous speed through each section of the lcd screen. An example of the dot movement can be found here.

3.9.5 Head-Tracking

The original VSS patch used an Oculus Rift as a head-tracking device which was also used to provide visuals of the VAE. As this project does not provide visuals, a smaller, less obtrusive device was sought. Initially the YEI 3-Space Sensor [35] was investigated, a small device that can track angular rotation and output results into Max. It was found however that the software used to interface between the device and the computer (a Mac) was only in a Beta stage. As a consequence, the output from the sensor did not function correctly or provide useful information.

Instead, the c74 application already being used for the iPad user interface (see section Location Selection) was used to extract rotational data from an iPhone. The iPhone data retrieved was accurate and provided a steady wireless connection, as opposed to the USB connection required

Figure 54: Image of an iPhone wrapped up in a hat used as a low-tech head-tracking device.

for the YEI sensor. For user testing, a low-tech head-mounting solution was found by wrapping the iPhone in the front of a cotton hat, shown in figure 54. This was integrated with the existing head-tracking system by creating a new patch that could the data from the iPhone (much like the patch used for the iPad) and map its rotational data to an angle from 0° to 360°, including a calibration system that ensures that each user is facing the same way in the VAE before they start turning their head. The output of this was sent to the input of the existing head-tracking system where the Oculus rift data would have been sent and is used to pan between the four direction RIR files simulating the effect of turning the the VAE.

3.9.6 Settings Menu

Figure 55 shows the settings patch that can be used to determine which RIR grid to use by selecting the number 1-5 on the top left of the patch (green buttons) as well as the sample rate of the RIR files. The audio input settings can be used to change which audio input is used. The panning settings determine the speed at which the user is moved between RIR locations (though this was kept at the same rate the system timer was set, 2.5s).

3.9.7 Software Issues

Location approximation

One issue raised when implementing iteration 3 was the way in which the user is moved through the virtual space. As only two RIR locations are used at a time as opposed to using 4 RIR locations for interpolation, the user is moved in a 'zigzag' pattern (an example of 4 RIR locations is used here as it is the minimum number that can be used other than 2). Figure 56 shows two images illustrating how the user is moved through the virtual space using the final iteration of the software against the initial two iterations, where the curved blue line represents the path drawn by the user. The image on the left shows how 4 RIR locations would be used together to approximate

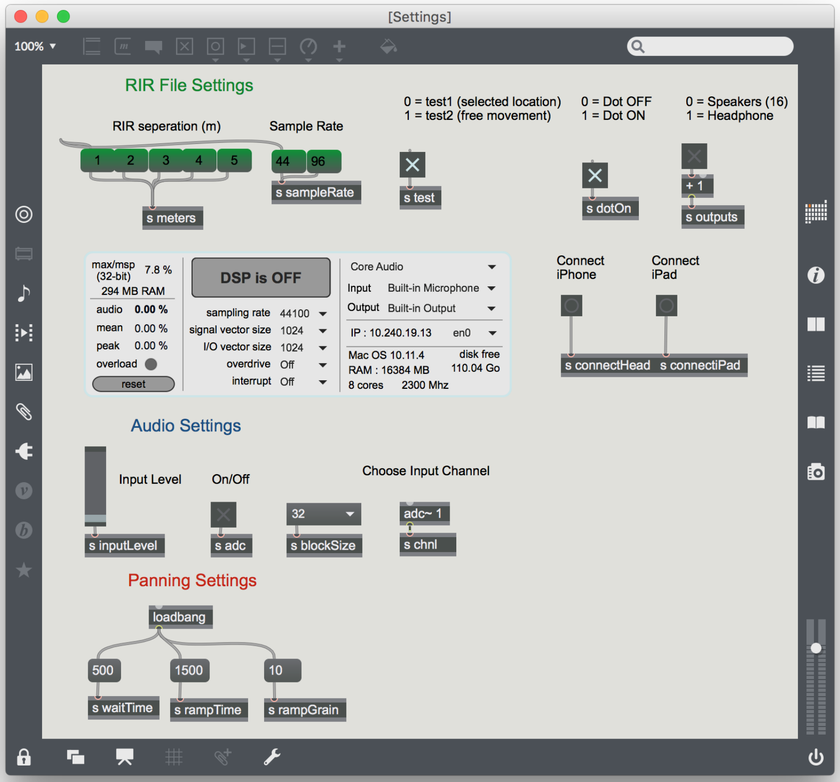


Figure 55: Settings patch used to control parts of the overall patch from one location

the room acoustics between them. The image on the right shows how the user is passed between locations one at a time, causing the 'zigzag' pattern.

Travel Speed

It has been mentioned that the rate at which RIR files are loaded into the system are determined by a timer that loops every 2.5 seconds. As a result of this, the rate at which the user is moved from one RIR position to the next is also 2.5 second, regardless of the distance between RIR location. This means than when the RIR locations are separated by a greater distance, the speed at which the user moves within the virtual space is also greater, creating an inconsistency in speed with each RIR grid used. This was not considered when the system was designed and as a consequence, negatively influenced the results obtained in user test #3 (see section Test #3).

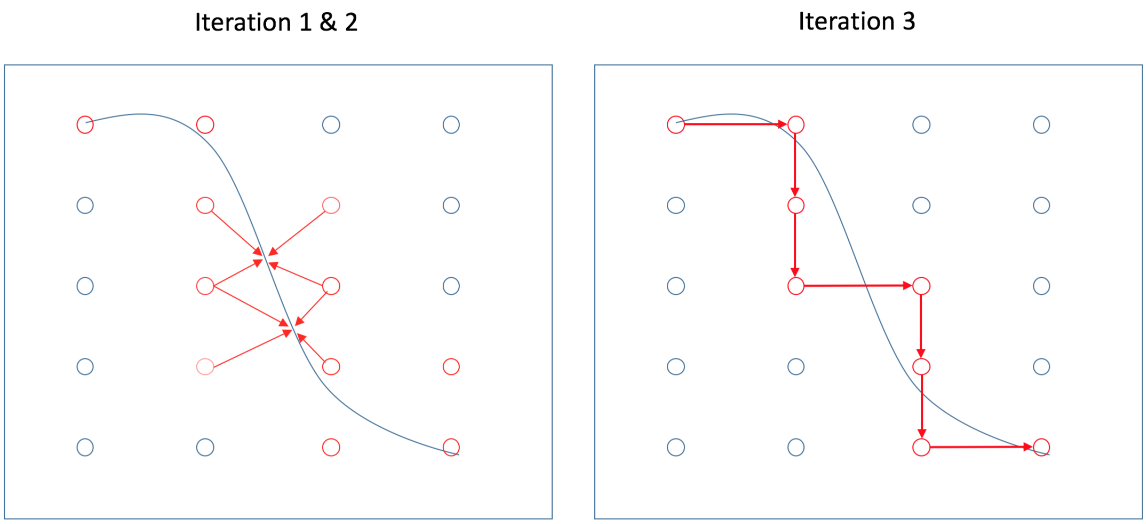


Figure 56: Illustration of how the user is moved through the virtual space using different numbers of RIR locations where the image on the **left** shows how 4 RIR's are used at a time to approximate the room acoustics of the unmeasured location and the image on the **right** shows the user being passed between two RIR locations at a time thus producing a 'zigzag' pattern. The blue line represents the paths drawn by the user and the circles represent RIR locations.

As previously explained in section Iteration 3 (Final), three convolution patches were used to move the user between two RIR locations while the third loads the next location. In an attempt to make the system less computationally expensive, if any of the convolution patched were loading a file as opposed to convolving an audio signal with one, the convolution algorithm was bypassed. However it was found that any delay in turning the bypass on or off produced a 'popping' noise. Small delays in turning the bypass on and off often occurred, therefore this bypass functionality was removed.

3.10 - Latency

The time taken for the input signal to run through the system causes the room reflections to arrive at the user at delayed, thus incorrect times. To determine the length of the delay, a latency test was conducted. The test procedure was similar to that in the production of the original VSS [1] with some slight differences. Two Earthwork M30 reference microphones [36] were placed in the centre of the speaker array. One was connected to the input of the Max patch and the other was connected to a Mac running reaper through a MOTU audio interface [37] to record a reference input. An output channel running to one of the Genelec speakers was routed to the second input of the MOTU interface. This was used to record the signal after it had run through the Max patch, thus allowing the two signals to be compared to find the latency time. Both systems were running at a sample rate of 44.1kHz. A pair of drumsticks was used to produce a sharp impulse, allowing a clear analysis of its onset time. This was then convolved with a 4-channel Dirac impulse (1

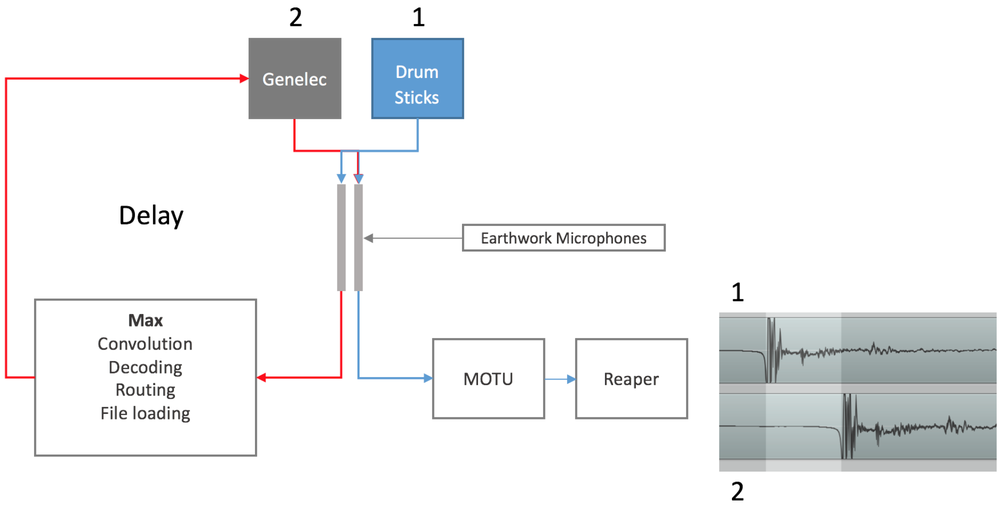


Figure 57: Diagram showing the latency measurement process. Red arrow shows the signal being delayed due to running through the max patch, eventually being output from the Genelec and running into reaper. Blue arrow indicates the direct signal to Reaper. The numbers indicate which of the audio waves comes from which source, **1** being the direct sound and **2** being the delayed sound.

sample of amplitude 1) of approximately 3 seconds long, meaning that the Max patch would run the same process as it usually would, but the resulting audio would not be reverberant, thus making the analysis easier. Figure 57 shows a diagram of the latency measurement process.

Previous testing of the system had shown that the system ran smoothly when using an input block size of 32 in Max, thus having a total latency of 22ms, however for future records (and in case the system speed were to slow down due to potential future changes) the latency was measured for varying block sizes. The following table shows the resulting latency time due to the different block sizes used in Max. The total latency is calculated by adding 5ms to the system latency which accounts for the time taken to travel from the loudspeaker to the centre of the speaker array where the user will be present.

Sample Rate (kHz)	Block Size	System Latency (ms)	Total Latency (ms)
44.1	512	27	32
44.1	256	20	25
44.1	128	18	23
44.1	64	17	22
44.1	32	17	22

3.11 - RIR Trimming

Due to the latency present in the system, the first 22ms of the RIR's had to be trimmed. Figure 58 shows two of the synthetic RIR's, the top showing one from the centre of the room (8.85m from the left wall) and the bottom showing an RIR taken 1.85m away from the left wall. Both of the RIR's being analysed were produced with the sound source facing the left wall (90°) to emphasis the early wall reflection, used to explain the issue caused due to the system latency.

Points (1), (2) and (3) are identical in both RIR's showing the start of the impulse (1), the direct sound from the source to the receiver (2) and the floor reflection (3). The main difference in the RIR's are the direct reflection times from the left wall. (4) shows the wall reflection occurring at 0.04084s, indicating a travel distance of 3.75m (following the same path described in figure 29 in section Results). (5) shows the reflection from the same wall, however occurring much later due to the greater distance from the wall. The point at which the RIR's will be trimmed is indicated by the red intersecting line, occurring at approximately 0.052s. It can be seen that the direct reflection from the left wall (4) is removed from the impulse, whereas in the RIR in the centre of the room, the direct wall reflection (5) remains in the trimmed version of the RIR. Due to the 22ms delay, any sound that travels less than 7.568m (if the source and receiver were placed 3.784 away from a surface) before reaching the receiver will be removed from the RIR.

The necessity for direct sound has been previously discussed in [38], stating that as reverb builds up over time eventually masking reflections, the direct sounds are used for sound source localisation. As the user in the VAE is both the sound source and the receiver, they will rely on direct reflections from the walls to determine their location within the space, however, the direct reflections from walls have had to be removed in some cases. Therefore, when moving closer to a wall, it may become difficult for the user to determine their locations.

The Matlab script used from audio file trimming can be found: RIRtrim.m or in file 2.

3.12 - Software Implementation Summary

Revisiting the aims regarding software implementation set in Project Aims and Motivation, it can be said that two out of the three accomplished:

1) The production of a user interface that can be used remotely from within the spherical speaker array.

2) The extension of the Max patch that could accommodate the user interface and load the appropriate RIR files required to place the user in the desired location within the VAE.

These has been achieved by producing the user interface using the 'c74' object, allowing the iPad to communicate with the Max patch and visa versa. The interaction with the user interface is interpreted by the 'UI_to_file' patch and the 'loadFilesLogic' javascript object. These successfully interpret the users intention to move to a position within the VAE and calculate the appropriate files to load.

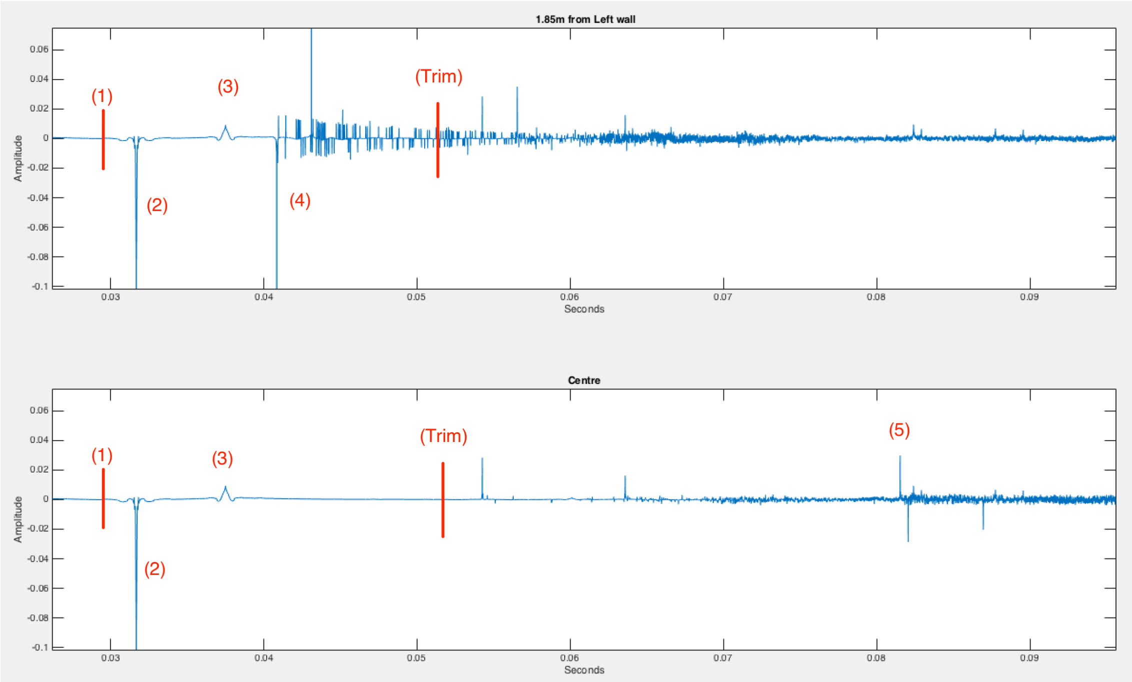


Figure 58

3) The production of a system that can interpolate between the appropriate RIR positions.

For the third aim set, it must be said that this was only partially fulfilled. Though the final iteration of the software (Iteration 3 (Final)) interpolates between the appropriate RIR's for the given design, it does not work the way originally intended due to the technological restrictions previously stated. Instead of allowing the user to 'freely' move around the space, they are instead restricted to the available RIR locations and are only able to move between them when moving from one set location to another, as opposed to being able to hover between them as would have been able using iteration 1 (Iteration 1).

However, as the user interface interpretation section of the software simple outputs the required file names, it can be integrated with an improved version of the mobility section in the future. Therefore with simplifications to the first iteration of the software, such as providing the user with a grid of four RIR locations instead of a full grid of 9, though less accurate, would be much quicker and would provide more freedom than the currently implemented version.

Due to this being a real time system, the latency introduced greatly effects a large number of the RIR files, as the start of file have to be trimmed by 22ms. This unfortunately removes some of the direct wall reflections for surfaces that are closer than 3.75m thus potentially effecting the plausibility of the system.

USER TESTING

Once the implementation of the system to allow the user to move themselves around a virtual space had been completed, the plausibility of the system was tested and the questions raised as to the effect on the perception of motion with the VAE when using this system were investigated. The following sections present the aims of the user tests in full, the procedure required to fulfil the set objectives followed by the results and a discussion.

In total, seven students took part in the user tests, all of which were studying audio related subjects.

4.1 - User Testing Aims

Test #1	Investigate the effect of using the synthetic RIR's as opposed to the previously used real RIR's.
Test #2	Investigate how far the user has to be moved before they notice that they have been moved using synthetic RIR's to investigate the potential minimum distance between synthetic RIR's required.
Test #3.1	Investigate which RIR grid provides the user with the best sense of mobility.
Test #3.2	Investigate whether the users perception differs when using a position feedback system.

Before the tests were conducted, each participant was presented with a 'Test Participant Form', informing them of the aims of the tests and the procedures that were to follow. As they were going to be stood inside the speaker array, the answers provided by each participant were taken down on their behalf. At the end of the experiment the answers were checked and signed by the participant assuring that their answers had been taken down accurately. The form provided to the participants along with the questions they were asked can be found in Appendix D.

Prior each test, the participants were allowed two 'dummy runs', allowing them to get used to the system without having their answers recorded, with the intention of removing the disadvantage of being unfamiliar with the system for the first few questions of each test. They were also informed that during each test they were allowed to turn their head in the space to obtain a more natural listening experience.

4.2 - Test #1

4.2.1 Procedure

To investigate the difference in the perception of distance moved when using either synthetic or real RIR's, the following procedure was carried out,

The participants were told that two methods were going to be investigated during this test, method **A** and method **B**. The participants were not told that in method **A**, the real RIR measurements from Hendrix Hall were going to be used to move them around the room and in method **B** the synthetic RIR's produced in Odeon were going to be used. The participants were told that using method **A**, they were going to be placed somewhere in the virtual space and that they would be asked to say the word 'Bob'. They would then be moved to another position in the space and asked to repeat themselves. This procedure would the be repeated, however this time using method **B**. They were then asked to state whether when method **B** was used, whether they felt they had:

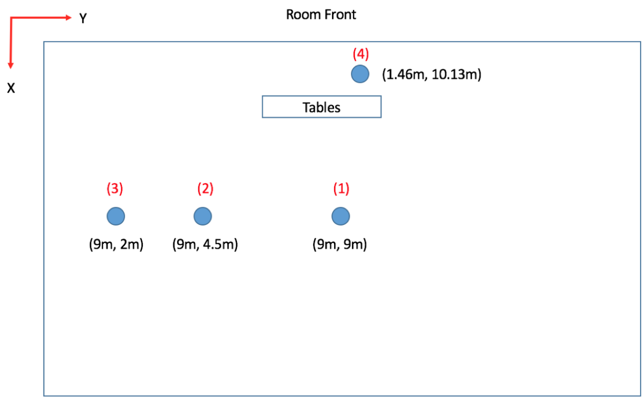
1) Moved a **shorter** distance than they had in **A**
2) Moved the **same** distance they had in **A**
3) Moved a **further** distance than they had in **A**
4) I don't know

This was repeated three times where the distances used for methods A and B were kept the same as each other. For the final two trials, the distances between the two methods were changed. The left of figure 59 shows an illustration of the virtual space indicating where the four RIR locations used for the test were. The black numbers indicate the coordinates of the RIR's relative to the top left corner of the room (note that the x and y axis are opposite to convention due to the way the building was modelled in Google SketchUp) and the red numbers indicate which location the user was moved to during each test, shown on the right in figure 59.

4.2.2 Results

The participants answers for user test #1 were taken and averaged across all five trials. Figure 65 shows these results, showing the percentage of correct and incorrect answers given (left) and the percentage of each type of answer given (right). It can be seen that only 23% of the answers given across all trials and participants were correct, indicating there being great difficulty comparing the two distances moved.

The pie chart on the right shows that more commonly, people thought that they were moving a further distance when the synthetic RIR's were used. It can be seen in the table on the right in figure 59 that the expected answers are as follows:



Trail	Real RIR **(A)**		Odeon RIR **(B)**	
	Start	End	Start	End
1	(1)	(2)	(1)	(2)
2	(1)	(3)	(1)	(3)
3	(1)	(4)	(1)	(4)
4	(1)	(3)	(1)	(4)
5	(4)	(1)	(1)	(3)

Figure 59: **Left:** Illustration of the RIR locations used in user test #1. **Right:** Table showing the pairs of positions the participant was moved to, corresponding to the positions shown in the diagram on the left.

Trial	**Correct Answer**
1-3	Same
4	Shorter
5	Further

However, an interesting result can be seen when breaking down the answers given into their respective trials, as shown by the three pie charts in figure 61. When the distance moved was either the same (trials 1-3) or shorter (trial 4), more people tend to answer that they had moved further, more so when they had actually moved a shorter distance. When the distance moved was actually further, the answers are evenly split between 'Further', 'Same' and 'Don't Know'. This negative correlation between expected answer and given answer suggests that the pie chart on the right in figure 65, showing that the majority of people thought they were moving further throughout all trials, is not influence by the fact that people got the answer to trial 5 correct, indicating that it is the use of the synthetic RIR's themselves that cause this perception.

4.2.3 Discussion

Due to the small percentage of correct answers given, it is apparent that comparing the difference in distance moved is not easy. For the system used, there may be 2 reasons for this:

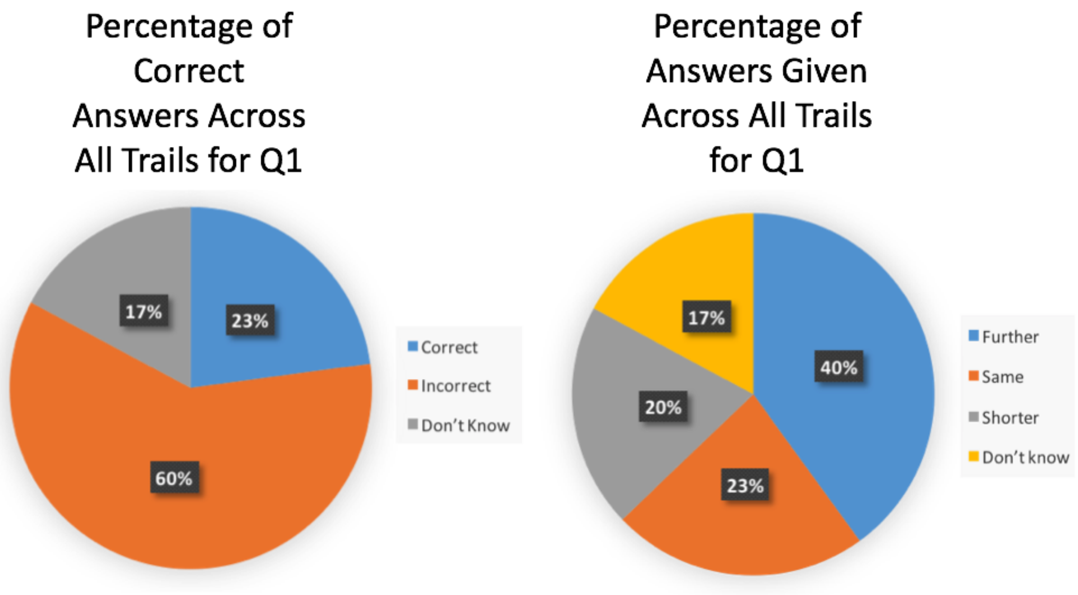


Figure 60: Pie charts showing the results of user test #1

1) **The test is too hard**
   Asking the participants to not only compare how two different locations sound to each other, but compare that difference to how another two sound compare to each other may have been too much to remember at once. This may have led to participants guessing answers as opposed to giving an answer based on opinion, even though the 'I Don't Know' option was available.
2) **Inaccurate RIR's**
   It has been mentioned in section RIR Trimming that due the fact that the RIR's had to be trimmed due to system latency, much needed direct wall reflections are not present for surfaces less than approximately 3.78m away, thus affecting the ability to accurately assess ones location within a space.

4.3 - Test #2

As this test was to obtain objective results, a hypothesis was formed: The participants would not notice they have moved for the first few trials given the short distance travelled. As the distance travelled increases, it is thought that the majority of participants would notice movement, producing a linear correlation between distance moved and 'Yes' answers given.

4.3.1 Procedure

The second user test is similar to that of the first test, though a little more simple. Figure 62 shows 8 RIR locations, where (1) is in the centre of the room. The participant was placed at position (1) and asked to say the word 'Bob'. They would then be moved closer to the left wall, asked

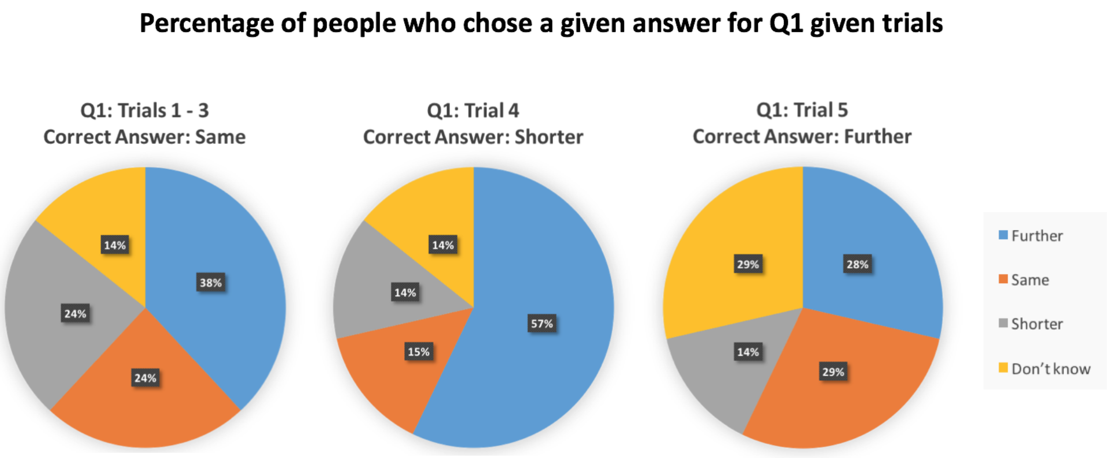
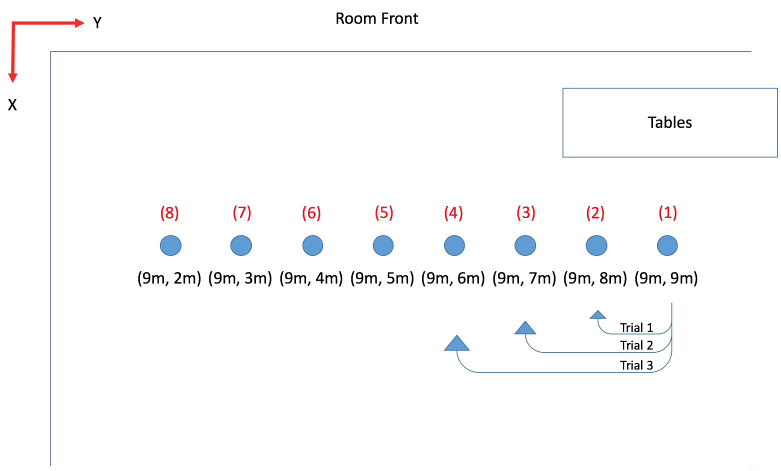


Figure 61: Three pie charts showing the percentage of answers given for: **Left:** Trials 1- 3, **Centre:** Trial 4, **Right:** Trial 5.

to repeat themselves and asked whether they thought they had moved or not, simply answering 'Yes', 'No' or 'I Don't Know'. This was done 7 times, moving the participant a further distance each time. The table on the right in figure 62 shows the start and end position for each trial which is illustrated on the left.



Trail	Position (x,y)	
	Start	End
1	(1)	(2)
2	(1)	(3)
3	(1)	(4)
4	(1)	(5)
5	(1)	(6)
6	(1)	(7)
7	(1)	(8)

Figure 62: **Left:** Illustration of the RIR locations used in user test #2. **Right:** Table showing the pairs of positions the participant was moved to, corresponding to the positions shown in the diagram on the left.

4.3.2 Results

Figure 63 shows a line graph of the results obtained from user test #2, showing on how many participants answered 'Yes', 'No' or 'Don't Know' for each of the trials.

The original hypothesis to this test was that the participants were unlikely to notice that they had moved for the first few trials as they were being moved such a short distance (1m, 2m), with the expectation that they would notice when they were being moved much larger distances, hence the reason there were moved 1m closer to the wall in each trial. However, it was found that all participants answered 'Yes' for the first trial (movement of 1m towards the left wall) and all other trials resulted in a number of participants giving an answer other than 'Yes'. Instead of a positive correlation between distance moved and 'Yes' answers given, there appears to be a dip from trials 2-4.

Figure 64 shows a compressed version of the data in figure 63, clearly indicating that at no point during the test, do any of the participants agree whether they have definitively moved or not. This indicates that the perception of mobility may differ from person to person. This could potentially be based on their listening experience, or possibly due to the consistency of the input signal (saying the word "Bob") being different between participants.

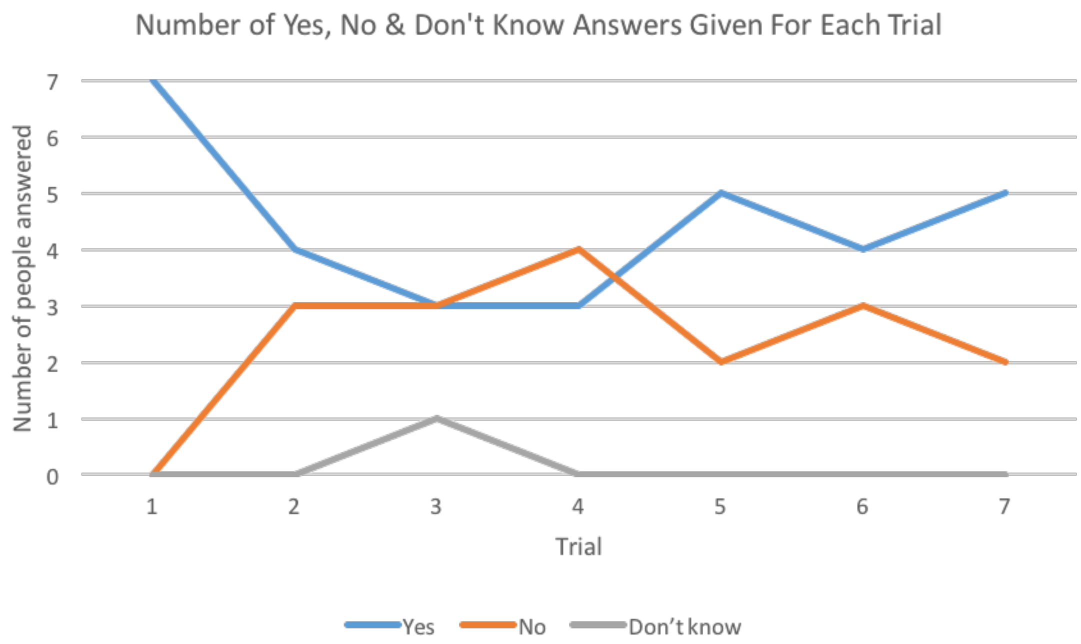


Figure 63: Line graph of the results from user test #2 showing the number of participants who answered 'Yes', 'No' or 'Don't Know' for each of the trials.

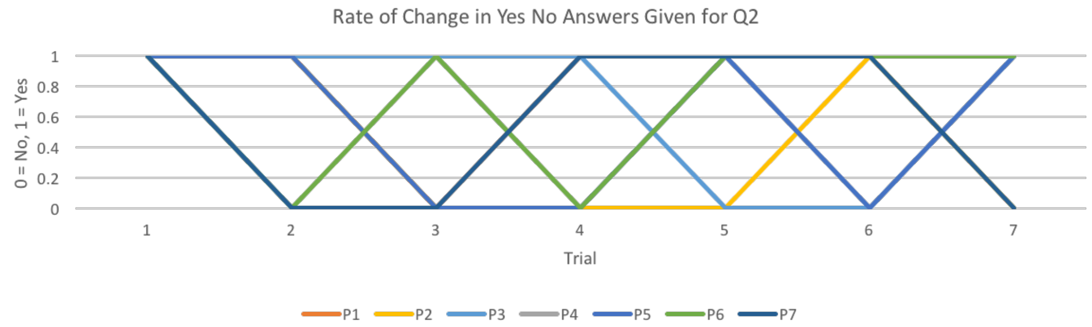


Figure 64: Line graph showing the rate at which participants give 'Yes' or 'No' answers, showing that at no point do all participants agree whether they have definitively moved or not. (The single 'Don't No' answer has been omitted here)

4.3.3 Discussion

Comparing the results to the original hypothesis, excluding the first two trails, there seems to be more people noticing movement as the distance moved increases thus showing a weak correlation between distance moved and 'Yes' answers given. These first two trials may have been subject to a bias, where the user is expecting to be moved. This however can not be backed up and remains a speculation. The fact that some participants could still not tell that they had moved at greater distances indicates that the differences between these RIR's may not be enough to convince some people that they have moved. A potential variable that may have caused some unexpected answers is the fact that no person can repeat themselves perfectly thus their perception of the location within the space is different each time. Another possible cause may be the lack of direct wall reflections due to the necessary RIR trimming, making it more difficult for participants to assess their location.

4.4 - Test #3

4.4.1 Procedure

The participants were presented with an iPad and informed that it could be used to draw a path within the virtual space which they would then be moved around. Five trials were run, each using a different RIR grid, previously explained in section Producing Different RIR Grids and displayed in section RIR Locations in figure 31 (1m separation) and in Appendix B as figures 72, 73, 74, 75 (2m - 5m separation respectively). The following table shows which trial used which RIR grid, stating the distance between RIR locations and the number of RIR locations within that grid:

Trial	RIR Location Separation	Number of Positions Available
1	1m	240
2	2m	112
3	3m	25
4	4m	12
5	5m	9

In each trial, the participant was asked to draw a path and listen to an audio sample [39] (`Trumpet.wav` which can be found here), which was played in place of the participant speaking/singing to provide a constant sound source (as determining movement would be difficult with inconsistent speaking). They were then asked to rate on a scale of 1-10 the quality of mobility, where 1 = extremely "jumpy" movement and 10 = completely smooth movement, or to give the answer 'N/A' if they had difficulty telling whether they were moving or not.

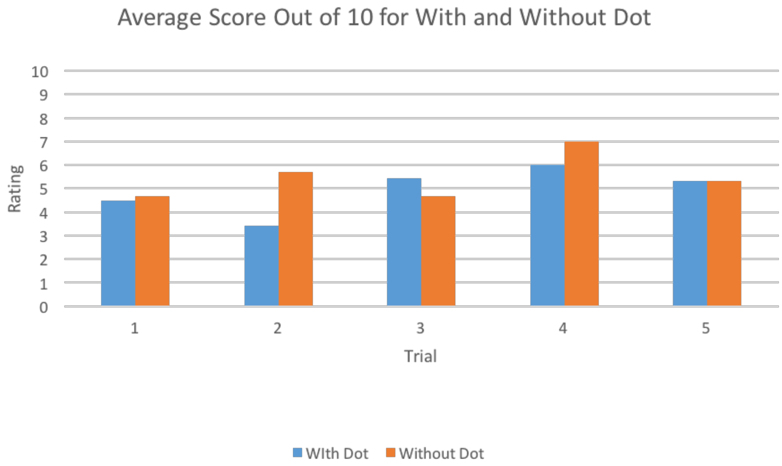
The five trials were run twice, once with the location tracking system described in Location Tracking turned off and once with it turned on, showing them where they were in the virtual space. This was to test whether their perception of how 'smoothly' they were moving around the space changed when they could see exactly where they were.

4.4.2 Results

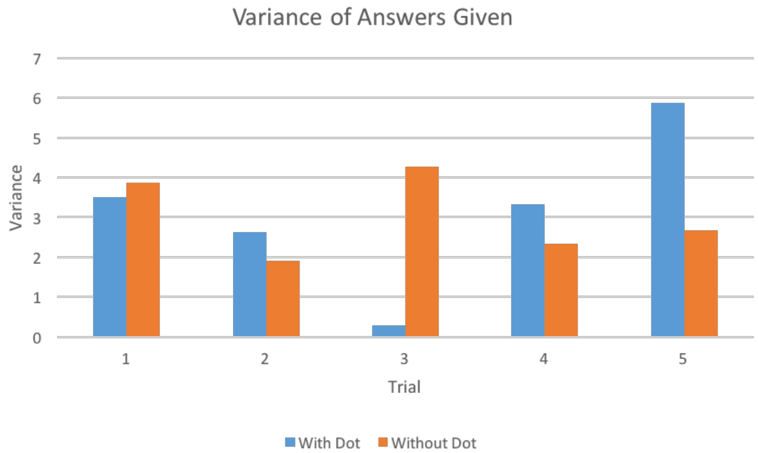
The results in figure 65a show that on average, using the grid with RIR locations separated by 4m when the location tracking dot was not used (blue bars), scored the highest. However it only scored a rating 0.5 higher than the grid using 3m of separation. By looking at that variance of the answers given in figure 65b, it can be seen that the grid with 3m separation was rated much more consistently that the grid with 4m separation.

When the location tracking dot was used, it was found the using the grid with 4m separation on average scored the highest with only a slightly higher variance than the grid of 2m separation, which scored second highest.

Participants were asked to provide comments on using the system both with and without the location tracking dot. Several participants commented on trial number 3, mentioning that it was easier to tell that they were moving than the prior trials. Comments on using the location tracking dot referred to the fact that it helped them determine what they should be listening for, one participant in particular referring to the speed the dot was moving at was important, giving them an indication as to how fast they should hear themselves moving given its variance across trials.



(a) Average score given from 1 - 10 (1 = jumpy movements, 10 = smooth movement) for each of the trials 1 - 5 (1 = 1m RIR separation, 5 = 5m RIR separation)



(b) Variance of the answers given in figure 65a

Figure 65: Bar chats showing the results of user test #3

4.4.3 Discussion

As discussed in section Iteration 3 (Final), the rate at which the user is moved from one RIR location to another is determined by a timer which runs at a constant speed. This means that when the distance between RIR locations is greater, the speed at which the participant is moved across the room is greater. The inconsistency in movement speed had an obvious effect on the perception of movement when using different sized RIR grids, where participants had commented that it felt like they were moving faster. For example, it had been noted by some participants that in trials 1 (1m RIR separation), it did not feel like they were moving. This is most likely because moving 1m every 2.5 seconds is too slow to notice. However, due to the compromise that had to be made due to system speed, they could not be moved any faster than this without the system failing to operate smoothly. This may have led to participants giving the trials that used more spacious RIR placement a higher score (such as trial 3 and 4) as they could hear themselves moving more obviously due to the greater distance travelled, thus thinking that they were moving more smoothly.

Due to the issues mentioned, the author concludes that these results cannot be used to confidently suggest how many RIR's are required to convince the user can move freely in the VAE and are therefore inconclusive. In order to obtain results regarding the original question, either the current system should be improved to keep the speed of the user constant, or a way to implement the original system, iteration 1, should be investigated.

4.5 - User Tests Discussion and Summary

4.5.1 Test Length

In total, all three user tests took around 45 - 60 minutes for each participant. Ideally, some of the test would have been longer allowing for a greater number of results to be gathered thus providing a more statistically reliable set of results. For example, the results for test #2 would benefit by making the user take the same test multiple times to check that variance in their answers in an attempt to minimise outliers, such as the unexpected in trial 1 (shown in figure 63). However, given that the user tests were already long, it was decided to not extend it any further. A possible solution to this would have been to do the three user tests separately at separate times, each one of them extended to obtain more reliable results.

4.5.2 The 'I Don't Know' Problem

Previous papers discussing the pros and cons of including an 'I Don't Know' (DK) answer, such as [40], suggest that including a DK response could leave holes in data by preventing a participant from having to do any cognitive work thus not coming to a conclusion or formulating an opinion for an answer, even if they have one, or it could prevent noise in the data due to participants having to provide a guess for an answer if they truly have no answer to give. Due to the difficult of these listening tests, it was decided that it was necessary to include a DK response, running the risk that the difficulty of the test may result in participants taking the easy option. However, during the user tests it was noted by the author that participants seemed hesitant about giving a DK answer, even when encouraged to do so when they were clearly struggling to formulate an answer. A lot of the time it was noticed that the answers given by participants were not confident. Though the analysis of the confidence in their answers given are informal and is simply the opinion of the author, it is felt that it is important to take into account for two reasons;

1) The small number of people who took part in the user tests.
2) The difficulty of the tests.

As there were only 7 people who took part in the test, a guess answer from any of the participants would heavily influence the obtained results. As it is thought that most of the participants were not sure of their answers, the obtained results are in fact unreliable. Coupled with the fact that the tests (especially test #1) were quite difficult, it is expected that guessing will have occurred. Again, these claims can not be backed up as the confidence in the participants answers are not formally part of this study, however it may explain some randomness in the results and should be kept in mind if further experimentation is to be done.

4.5.3 Results Summary

Test #1

When using the synthetic RIR's, participants were more likely to feel like they had moved further than when using the real RIR's regardless of the difference in distance moved.

Test #2

Results show that there is a weak correlation between the distance moved and the number of people who notice they have moved. However it has been shown that at no point do all the participants agree on whether they have moved or not, showing a difference in perception between people regardless of distance moved.

Test #3

For the currently implemented system, using an RIR grid with 3m separation provides the best experience for free mobility around the virtual space.

Conclusion

5.1 - Project Summary

A digital model of Hendrix Hall was designed and used to produce a large grid of synthetic RIR's in order to implement a direct RIR rendering system with the aim of allowing a user to move themselves around a virtual space. The original Max patch designed for the VSS was built upon to accommodate this new functionality by implementing an automated file loading system and a RIR interpolation system that approximates the movement defined by the user. Three user tests were carried out to access the plausibility of the system as a whole and to uncover information regarding the perception of mobility with regards to the implemented system.

5.2 - Project Success Evaluation

To measure the success of the project, the aims set in Project Aims and Motivation can be evaluated as follows:

1) Was the desired direct RIR rendering system fully implemented?

Partially. Though the desired system was produced in Max (section Iteration 1), due to technological issues regarding the speed at which it ran, it could not be used for a real time system such as the one required. Instead, a compromise between desired functionality and reliability was made (section Iteration 3 (Final)), where instead of a user being able to place themselves between RIR locations, they were restricted to the strict locations available within the RIR grid. So, though technically the desired system was not used, the compromise still allowed the user to achieve a similar result.

Upon these grounds, the author believes this aim had been achieved partially successfully.

2) Were comprehensive results obtained regarding the number of RIR's required to produce a plausible system?

Partially. The results obtained from user test #3 give an indication to the optimal number of RIR's required when the speed of mobility in the system varies depending on RIR separation. However, the results do not indicate the minimum number of RIR's that can be used to convince a user that they can freely move around a virtual space, which was the original intention of this test. This was a result of overlooking some of the undesired functionality of the new system. However, results from user test #2 show that some of the participant perceive movement when moved a distance as short as 2m (excluding the outlier of trial 1, where all participants reported that they could tell they had moved) while there is a rise in the number of people who perceive movement after a distance of 4m. This result in combination with the results from user test #3, suggesting that movement becomes noticeable with an RIR of 3 - 4 meters, indicating that using RIR's separated by a distance less than this may not be necessary.

So, though useful results for the implemented system have been acquired, to investigate the minimum number of RIR's required for the originally intended system, further investigation is required, thus the author again believes that this aim had been achieved partially successfully.

3) Were comprehensive results obtained regarding the perceptual difference in moving varying distances within a the VAE when using synthetic RIR's as opposed to the real RIR's?

Yes. The results from user test #1 show that the majority of participants answered incorrectly when asked if they felt they had moved a distance that is shorter, the same or further than they had moved when they were placed in the VAE when using real RIR's. This indicates that the use of different RIR's may effect the user perception of mobility. More thorough results could have been obtained by testing the users ability to perceive the difference in distance moved when using two sets of the same RIR's (eg, using only synthetic RIR's and seeing if the user was more likely to get these answers correct).

5.3 - Overall System Implementation

As mentioned in Project Aims and Motivation, using the direct RIR rendering method to implement mobility is simple, however the time required to implement such as system from scratch has the potential to deter people from recreating it. Now that the system is in place however, the process of creating another VAE would simply require the modelling of the desired room. Though this was one of the most time consuming parts of the process, it is the only process that would need to be redone.

One of the issues raised during the project was the effect of the system latency on the accuracy of the RIR's. As the direct wall reflections were removed from a large portion of the RIR files the system became a lot less accurate. It is possible that this latency may come down to the software used, Max. It may be possible to reduce the latency of the system by reproducing it in a more efficient language, such as C++.

Though it has been mentioned that the initially designed system was not able to be fully implemented, there now exists a system that can be easily built upon to implement different methods of virtual location simulation.

Rendering synthetic RIR's has shown to have its advantages over recording real RIR's such as being able to produce them in bulk without having to leave a computer. Also the option to move the source and receiver closer than typical methods of RIR recordings would allow (when using a loudspeaker and a microphone) and modelling a human head by using the appropriate directivity pattern was simple to do. However, it has been shown that issues such as the lack of necessary data for correctly modelling a room, such as the material list, has a significant impact on the quality of the produced RIR's, as shown in RIR Comparison.

5.4 - Final Comments

It was found that for a real time application such as the one required for the VSS, that the system implemented requires further work before it can be considered a beneficial tool for musical performers.

FURTHER WORK

6.1 - SOFTWARE

6.1.1 Storage

An issue with using the direct RIR rending method as stated in [18] is the storage required for the grid of RIR's. For this project, multiple grids were produced by taking the large grid of RIR's, extracting and storing the appropriate ones in separate files. This meant that a number of duplicate files were having to be stored. If time had allowed, a system in Max would have been designed to do the file extracting in real time, therefore requiring only one group of RIR files as opposed to five.

6.1.2 Iteration 1 Redesign

The failure to implement the desired RIR interpolation method in the software used in this project has been noted several times, mentioning the system speed as being the culprit. As this design is more desirable than the one currently implemented, further work on the original design would be encourage.

One possible improvement could be to simplify the grid used for interpolation. Figure 66 illustrates three implementations; **Grid 1** shows the layout of the original grid used where the user can move freely within the grid of RIRs. **Grid 2** shows a layout where only the RIR locations at 90 degree increments to the user are available reducing the area in which the user can move (shown by the grey square). **Grid 3** shows a similar layout but using the RIR locations at diagonals to the user. This would increase the area in which the user could move, however it would provide a less accurate approximation for the users position as the RIR's are more spread out.

Another possibility in producing a faster running system could be to redesign the software in a more efficient programming language such as C++. This also has the potential to reduce system latency, thus reducing the amount of time needed to be cut off the beginning of the RIR. This would in turn potentially avoid having to trim the direct wall reflections of a number of RIR files.

6.1.3 UI cross platform

As the design in the Max patch enables the user interface to be resized for any type of screen while still maintaining the correct ratio of RIR placements. It would be nice to see the user interface implemented on a less obtrusive device, allowing the user to change their location on their smartphone.

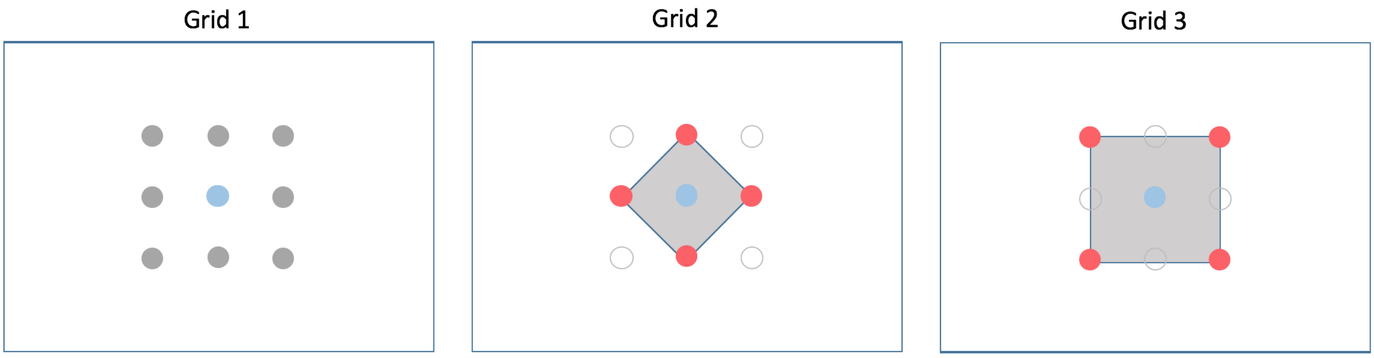


Figure 66: Illustration of how the grid of RIR positions used for interpolation could be simplified in an attempt to speed up the system. **Grid 1** shows the layout of the original grid used. **Grid 2** shows a layout where only the RIR locations at 90 degree angles to the user are available reducing the area in which the user can move (shown by the grey square). **Grid 3** uses RIR locations at diagonals to the user increasing the area available but decreasing accuracy.

6.2 - Accuracy

6.2.1 RIR Modelling

As mentioned in Project Aims and Motivation, using the direct RIR rending method allows for the calculated of RIR's offline. This lends it self to the potential of using RIR's calculated using the more accurate, but computationally expensive wave-based methods. Using methods discussed in [21], it is possible to combine the accurate low frequency models with the much less computationally expensive high frequency accurate models calculated using geometrical acoustic modelling methods used in Odeon. If this were done, it would be interesting to re-run user test #1 to see whether this makes a difference to the perception of distance moved with using the more accurate synthetic RIR's.

6.2.2 Surface Materials

One issue raised when modelling the room (Material Selection) was the lack of common materials provided in Odeons material list. This caused issues when trying to produce an RIR that more closely resembles the real measurements taken from Hendrix Hall. This could be improved by obtaining more accurate data regarding the rooms interior.

6.3 - Additional Functionality

The original implementation of the VSS used an Oculus rift as a head-tracking device and as a means to provide a 360° image of the position that the user was placed in. It was decided not to include this feature in this project as it would require obtaining images for each of the RIR locations. This would also have had an effect of the perception of mobility as it would not have been possible to convince the user they could move to any location they would like when they can see that they are being placed statically. It would be an interesting extension to include a system that provides the user with a graphical representation of the modelled space in real time, by using the model designed in Google SketchUp and investigating the difference in perception when this feature is active.

Project Management

7.1 - Management Methods

Upon starting the project, a simply Gantt chart was constructed to lay out the tasks that needed to be complete and the approximate time scale that should be allocated to each one. Very little of the project objectives changed from the initial planning other than one task which was producing low frequency accurate synthetic RIR's which would have taken a significant amount of time. This was removed from the project scope before the project implementation began and therefore was allocated no time.

Two weeks into the project it was decided that the Gantt chart was not an appropriate method for project planning. This is because some tasks (such as producing synthetic RIR's) relied solely on the completion of the preliminary tasks. Once it was discovered that room modelling would take a lot longer than originally thought due to inexperience with such as task, the Gantt chant would have had to be redesigned which would have required even more time. Instead, a series of check lists were used to keep track of which tasks needed completing. These tasks were written on post-it notes and used in a large Kanban board situated on the authors bedroom wall, seen in figure 67. This dynamic approach to project planning was much better suited to this project. Along with this, weekly emails were sent to the projects supervisor with an update regarding the progress of the project and any problems that were faces, as well as regular meetings with both the projects supervisor and a larger group of students with audio related projects.

Figure 77 in Appendix E shows the initial Gantt chart with a red line indicating at what point it was abandoned for a more appropriate planing method.

7.2 - Dealing With Issues

When issues arose in the project that were unexpected therefore consuming unallocated project time, time limits on fixing issues were set. An example of this is the software implementation. When the issue arose where the originally designed system (Iteration 1) did not work as intended, a compromise had to be found. However, as the majority of the time had been put into designing the original software patch, there was little time left to design the second. Due to this, the final implementation of the software does not work exactly as intended, however setting the time limit allowed for a system to be built that could be used for the user tests, as opposed to improving the system further but not being able to complete the user tests. It was found that sacrificing the quality of certain aspects in the project (such as the software) allowed for the completion of the project and a more thorough analysis of the project process as a whole.

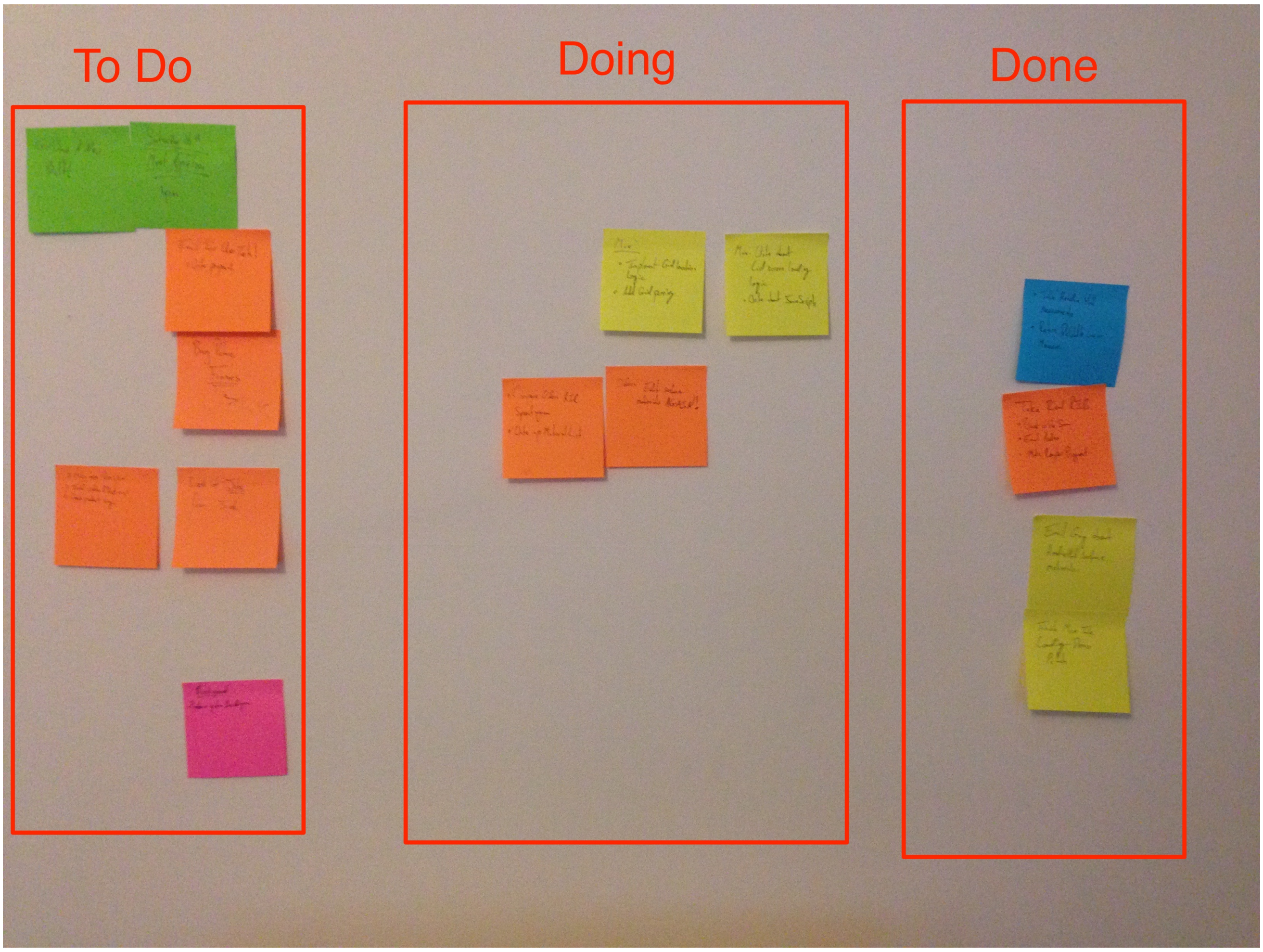


Figure 67: Image of the Kanban board used for keeping track of which tasks needed doing, which ones were ongoing and which tasks had been completed. This image was taken mid-way through the project.

REFERENCES

[1] J. S. Brereton and B. A. Hons, "Singing in space (s): singing performance in real and virtual acoustic environments — singers ' evaluation , performance analysis and listeners ' perception .," no. August, 2014.

[2] J. Huopaniemi, L. Savioja, T. Lokki, and R. Väänänen, "Virtual acoustics - applications and technology trends," *Proc. European Signal Proc. Conf. (Eusipco'2000)*, pp. 2201–2208, 2000.

[3] G. Stan, J. Embrechts, and D. Archambeau, "Comparison of different impulse response measurement techniques," *Journal of the Audio Engineering Society*, vol. 50, no. 4, pp. 249–262, 2002, ISSN: 1549-4950. [Online]. Available: http://orbi.ulg.ac.be/handle/2268/34825 (visited on Apr. 25, 2016).

[4] S. W. Smith, "The delta function and impulse response," *The Scientist and Engineer's Guide to Digital Signal Processing*, pp. 107–122, 2003.

[5] Iso 3382-1, "Acoustics - measurement of room acoustic parameters - part 1: performance spaces," vol. 3, 2009.

[6] H. Robjohns. (2001). You are surrounded, sos, [Online]. Available: http://www.soundonsound.com/sos/oct01/articles/surroundsound3.asp.

[7] P. J. Power, "Future spatial audio : subjective evaluation of 3d surround systems," PhD thesis, Acoustics Research Centre, University of Salford. [Online]. Available: http://usir.salford.ac.uk/34100/1/P%20Power_Thesis_Final.pdf (visited on May 7, 2016).

[8] P. White. (2016). Recording a live choral performance, sos, [Online]. Available: http://www.soundonsound.com/sos/jun04/articles/liveconcert.htm (visited on Apr. 25, 2016).

[9] S. Siltanen, T. Lokki, and L. Savioja, "Rays or waves? understanding the strengths and weaknesses of computational room acoustics modeling techniques," *Proceedings of the International Symposium on Room Acoustics, ISRA 2010*, no. August, pp. 1–6, 2010. [Online]. Available: http://www.acoustics.asn.au/conference%7B%5C_%7Dproceedings/ICA2010/cdrom-ISRA2010/Papers/O5a.pdf (visited on Apr. 25, 2016).

[10] (2016), Odeon, [Online]. Available: http://www.odeon.dk/pdf/ODEONManual.pdf (visited on Apr. 25, 2016).

[11] J. H. Rindel, "Computer simulation techniques for acoustical design of rooms," *Acoustics Australia*, vol. 23, pp. 81–86, 1995. [Online]. Available: http://www.odeon.dk/pdf/AustralAc_1995_Rindel.pdf (visited on Apr. 25, 2016).

[12] C. L. Christensen and G. Koutsouris, *Odeon user's manual*, Scion DTU Diplomvej Kgs. Lyngby Denmark, 2015. [Online]. Available: http://www.odeon.dk/pdf/ODEONManual12.pdf (visited on Apr. 25, 2016).

[13] Google. (2015). Google sketchup, [Online]. Available: http://www.sketchup.com/ (visited on Apr. 25, 2016).

[14] (2016). Su2odeon, Odeon, [Online]. Available: http://www.odeon.dk/su2odeon-plugin-trimble-sketchup (visited on Apr. 25, 2016).

[15] (2016). Max/msp, Cycling74, [Online]. Available: https://cycling74.com/products/max/ (visited on Apr. 25, 2016).

[16] (2016). Spat, IRCAM Centre Pompidou, [Online]. Available: http://forumnet.ircam.fr/product/spat-en/ (visited on Apr. 25, 2016).

[17] (2016), Oculus, [Online]. Available: https://www.oculus.com/en-us/ (visited on Apr. 25, 2016).

[18] L. Savioja, J. Huopaniemi, T. Lokki, and R. Väänänen, "Creating interactive virtual acoustic environments," *J. Audio Eng. Soc*, vol. 47, no. 9, pp. 675–705, 1999. [Online]. Available: http://www.aes.org/e-lib/browse.cfm?elib=12095 (visited on Apr. 25, 2016).

[19] R. Mehra, N. Raghuvanshi, M. Lin, and D. Manocha, "Efficient gpu-based solver for acoustic wave equation," pp. 1–10, 2010. [Online]. Available: http://mitran-lab.amath.unc.edu:8081/subversion/PetaFlopAcoustics/StochasticHelmholtz/biblio/Efficient%20GPU-Based%20Solver%20for%20Acoustic%20Wave%20Equation%20Nikunj%20Manocha.pdf.

[20] S. Oxnard and D. T. Murphy, "Room impulse response synthesis based on a 2d multi-plane fdtd hybrid acoustic model," *IEEEWorkshop on Applications of Signal Processing to Audio and Acoustics*, pp. 1–4, 2013.

[21] A. Southern, S. Siltanen, and L. Savioja, "Spatial room impulse responses with a hybrid modelling method,"

[22] *Genelec 8040b active monitoring system*, Data Sheet, Genelec, 2016. [Online]. Available: http://www.genelec.com/sites/default/files/media/Studio%20monitors/8000%20Series%20Studio%20Monitors/8040B/genelec_8040b_datasheet.pdf (visited on Apr. 25, 2016).

[23] (2016). St450 mkii soundfield portable, TSL Products, [Online]. Available: http://www.tslproducts.com/soundfield/st450-soundfield-portable/#spec-tab (visited on Apr. 25, 2016).

[24] (2016). Sine sweep matlab script, University of York, [Online]. Available: http://www.elec.york.ac.uk/internal_web/gsp/amt/modules/Audio_Acoustics_and_Applications/ (visited on Apr. 25, 2016).

[25] *Dewalt laser distance measurer manual*, Data Sheet, DeWalt, 2016. [Online]. Available: http://service.dewalt.co.uk/PDMSDocuments/EU/Docs//docpdf/dw03101-typ1_en.pdf (visited on Apr. 25, 2016).

[26] (2016). Absorption coefficients, AcousticTraffic, [Online]. Available: http://www.acoustic.ua/st/web_absorption_data_eng.pdf (visited on Apr. 25, 2016).

[27] (2016), BemBook, [Online]. Available: http://www.bembook.ibpsa.us/index.php?title=Absorption_Coefficient&action=edit (visited on Apr. 25, 2016).

[28] P. Kovalchik, R. Matetic, S. Peng, and G. Cole, "Measurement method for determining absorption coefficients for underground mines," pp. 1–4, [Online]. Available: http://www.cdc.gov/niosh/mining/userfiles/works/pdfs/ammfd.pdf (visited on Apr. 25, 2016).

[29] B. B. Monson, E. J. Hunter, and B. H. Story, "Horizontal directivity of low- and high-frequency energy in speech and singing.," *The Journal of the Acoustical Society of America*, vol. 132, no. 1, pp. 433–41, 2012, ISSN: 1520-8524. DOI: 10.1121/1.4725963.

[30] C. Armstrong, "A filter based approached to simulating acoustic radiation patters through a spherical array of loudspeakers," (Unpublished Masters Dissertation, University of York, May, 2016).

[31] L. Thresh, *Impulse response analysis question*, Personal Email, Odeon Support, Mar. 2016.

[32] A. F. Marina Theodoropoulou. (2016). Operatic voice, OpenAir, [Online]. Available: http://www.openairlib.net/anechoicdb?keys=singing&field_sound_cat_value=All (visited on May 5, 2016).

[33] (2016). Mira, Cycling74, [Online]. Available: https://cycling74.com/products/mira/ (visited on Apr. 25, 2016).

[34] (2016). C74, Cycling74, [Online]. Available: https://cycling74.com/project/project43-c74-iphone-app/#.Vv1FhRIrJE4 (visited on Apr. 25, 2016).

[35] Y. Technology, *3-space sensor embedded*, 630 Second Street, Portsmouth, Ohio 45662. [Online]. Available: http://www.solarsystemexpress.com/uploads/5/0/6/0/5060129/3-space_sensor_users_manual_embedded_1.1_r13.pdf (visited on May 2, 2016).

[36] *M30 high definition measurement microphone*, Data Sheet, Earthworks, 2016. [Online]. Available: http://www.earthworksaudio.com/wp-content/uploads/2016/03/M30-Data-Sheet-2016.pdf (visited on May 3, 2016).

[37] (2016). Motu ultralite-mk3, MOTU, [Online]. Available: http://motu.com/products/motuaudio/ultralite-mk3 (visited on May 3, 2016).

[38] S. Devore, A. Ihlefeld, K. Hancock, B. Shinn-Cunningham, and B. Delgutte, "Accurate sound localization in reverberant environments is mediated by robust encoding of spatial cues in the auditory midbrain," *Neuron*, vol. 62, no. 1, pp. 123–134, 2009, ISSN: 08966273. DOI: 10.1016/j.neuron.2009.02.018.

[39] C. Steele-Perkins. (2016). 'the trumpet shall sound' anechoic trumpet sample, OpenAir, [Online]. Available: http://www.openairlib.net/sites/default/files/anechoic/data/helena-daffern/handel-trumpet/mono/tr-1888-piece2-t-32.wav (visited on May 6, 2016).

[40] Paul Beatty & Douglas Herrmann, "A framework for evaluating "don't know" responses in surveys," *Vasa*, pp. 1005–1010, 1989. [Online]. Available: http://medcontent.metapress.com/index/A65RM03P4874243N.pdf.

Appendices

Appendix A

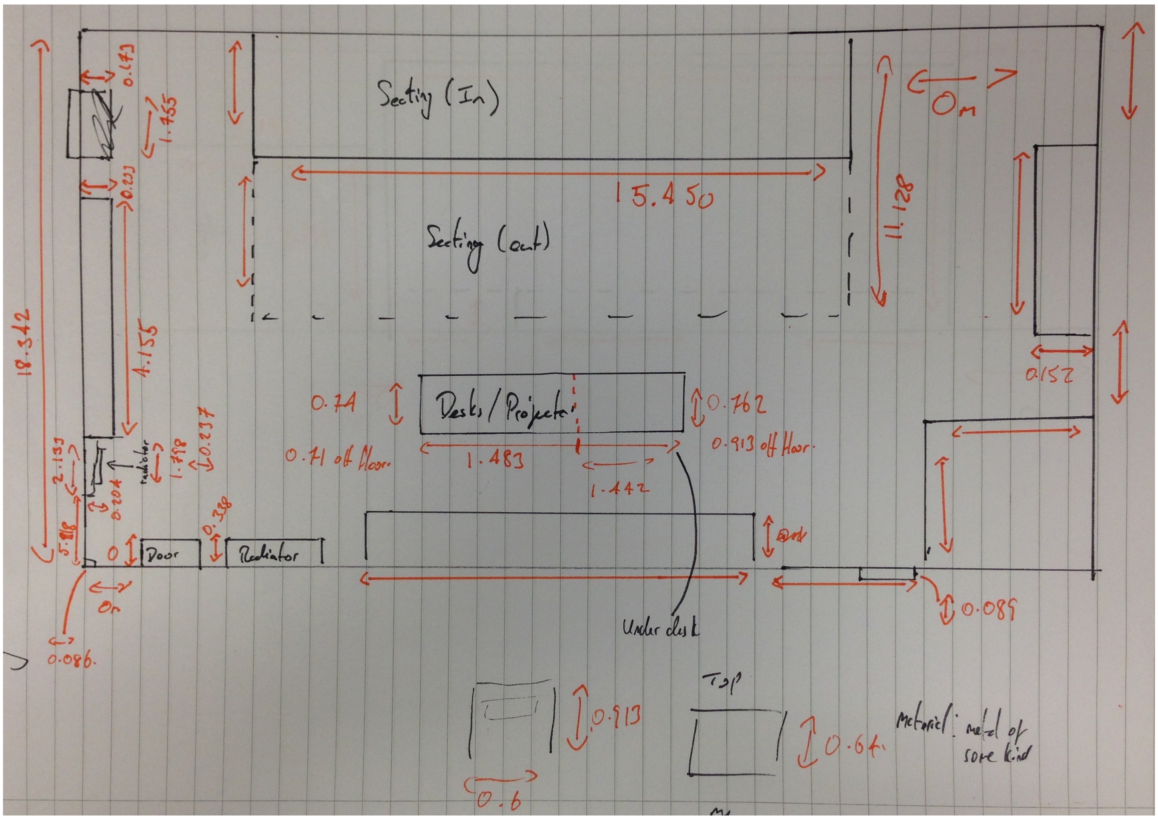


Figure 68: Annotated blueprint of Hendrix Hall from a birds-eye view

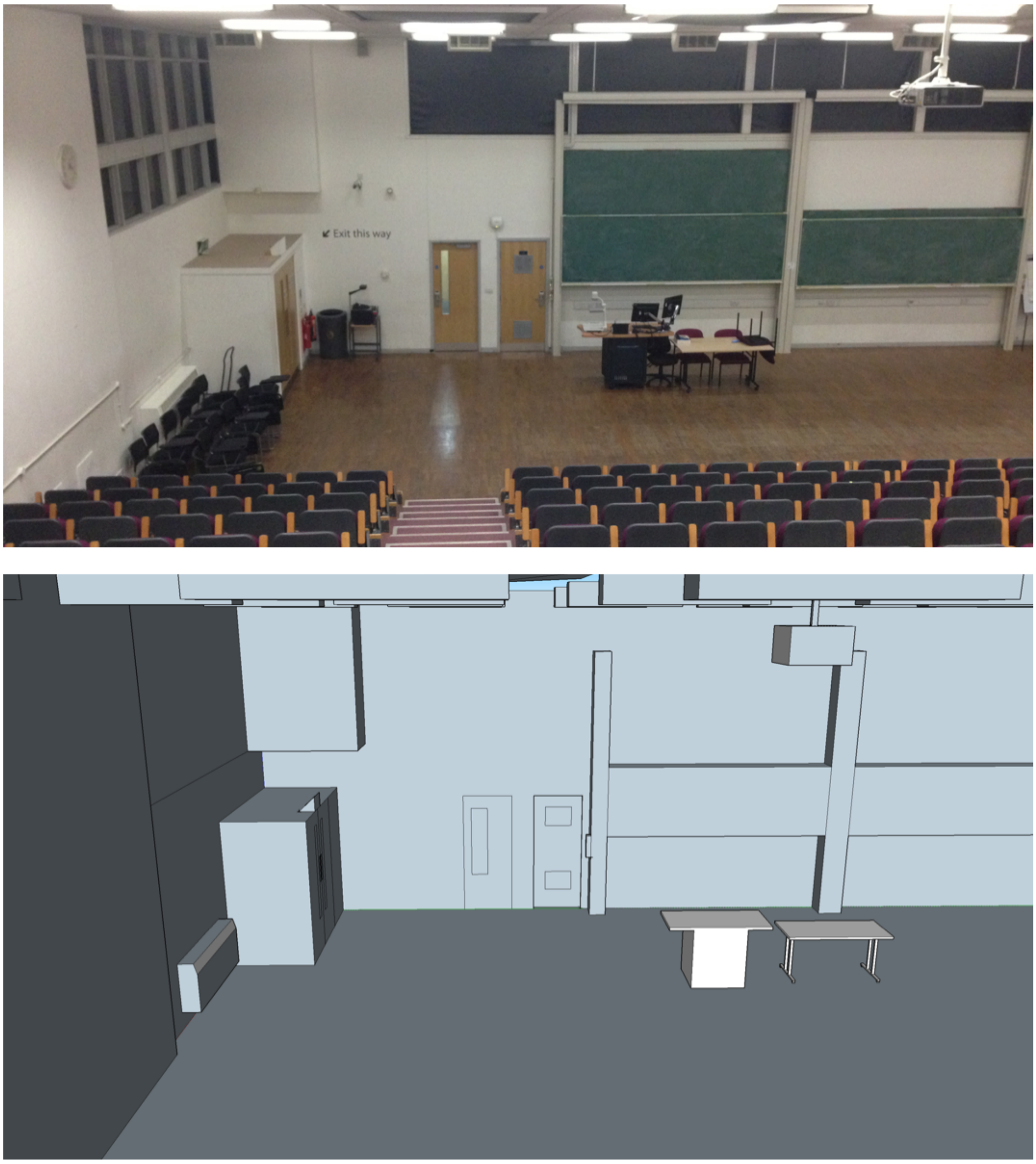


Figure 69: Image comparing a picture taken in Hendrix Hall (top) Compared to an image taken from the Google SetchUp Model (bottom)

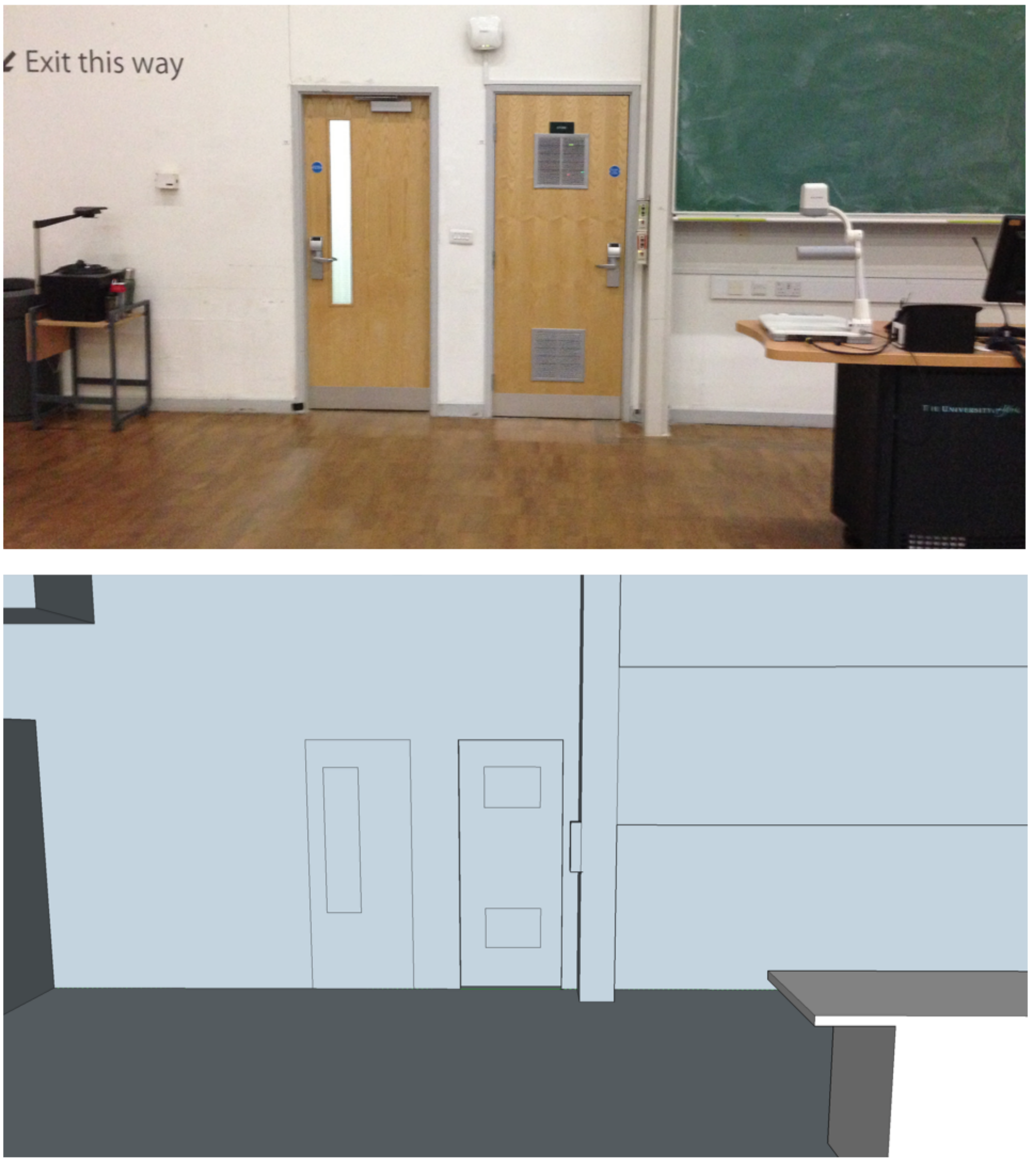


Figure 70: Image comparing a picture taken in Hendrix Hall (top) Compared to an image taken from the Google SetchUp Model (bottom)

Figure 71: Image comparing a picture taken in Hendrix Hall (top) Compared to an image taken from the Google SetchUp Model (bottom)

APPENDIX B

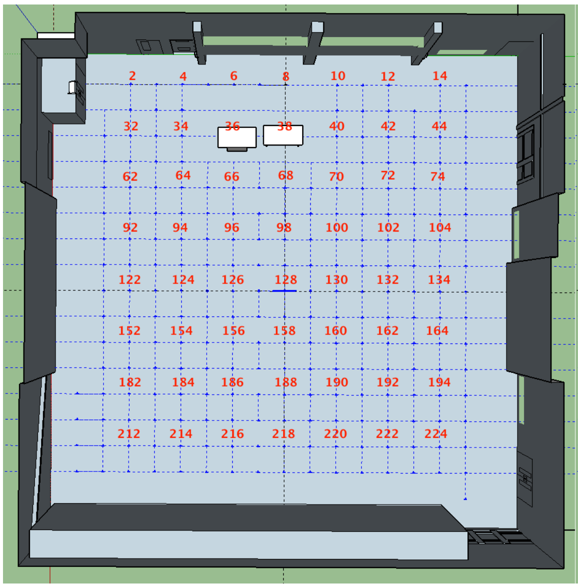


Figure 72: RIR grid with 2m separation

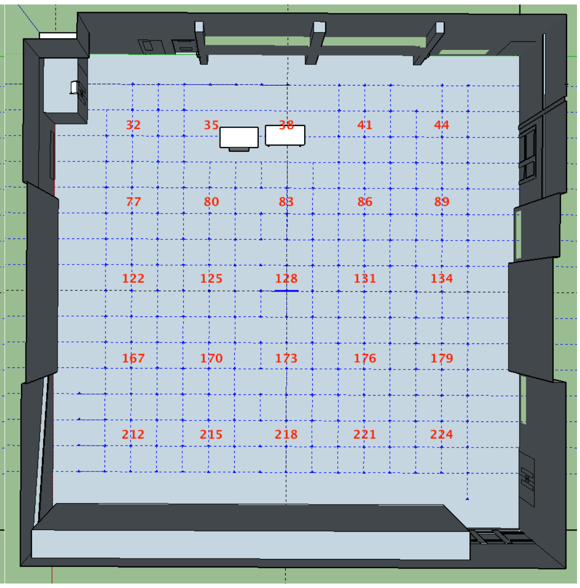


Figure 73: RIR grid with 3m separation

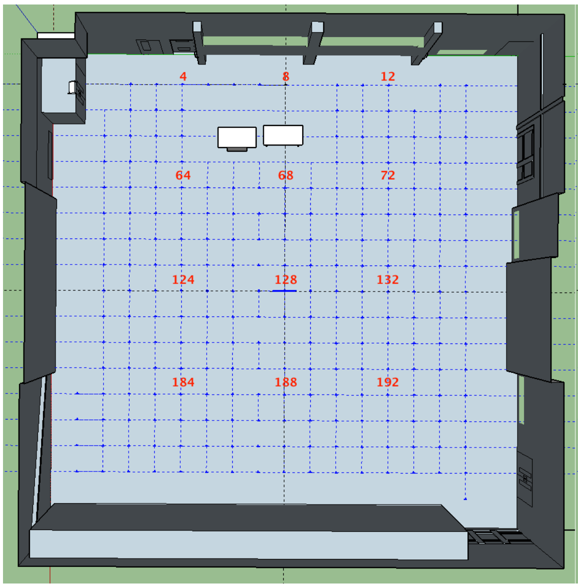


Figure 74: RIR grid with 4m separation

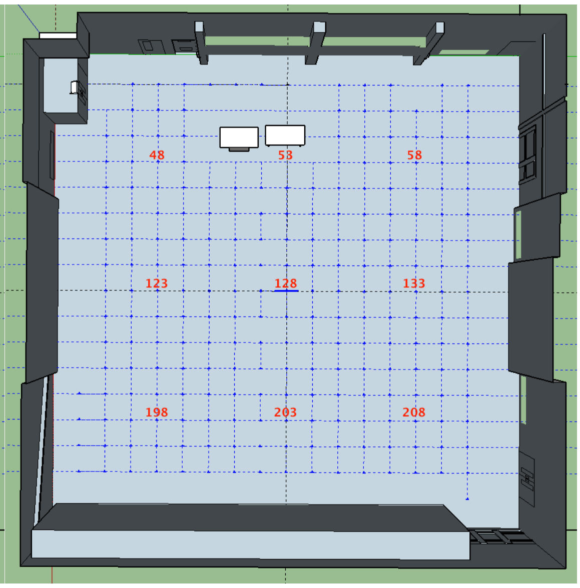


Figure 75: RIR grid with 5m separation

Appendix C

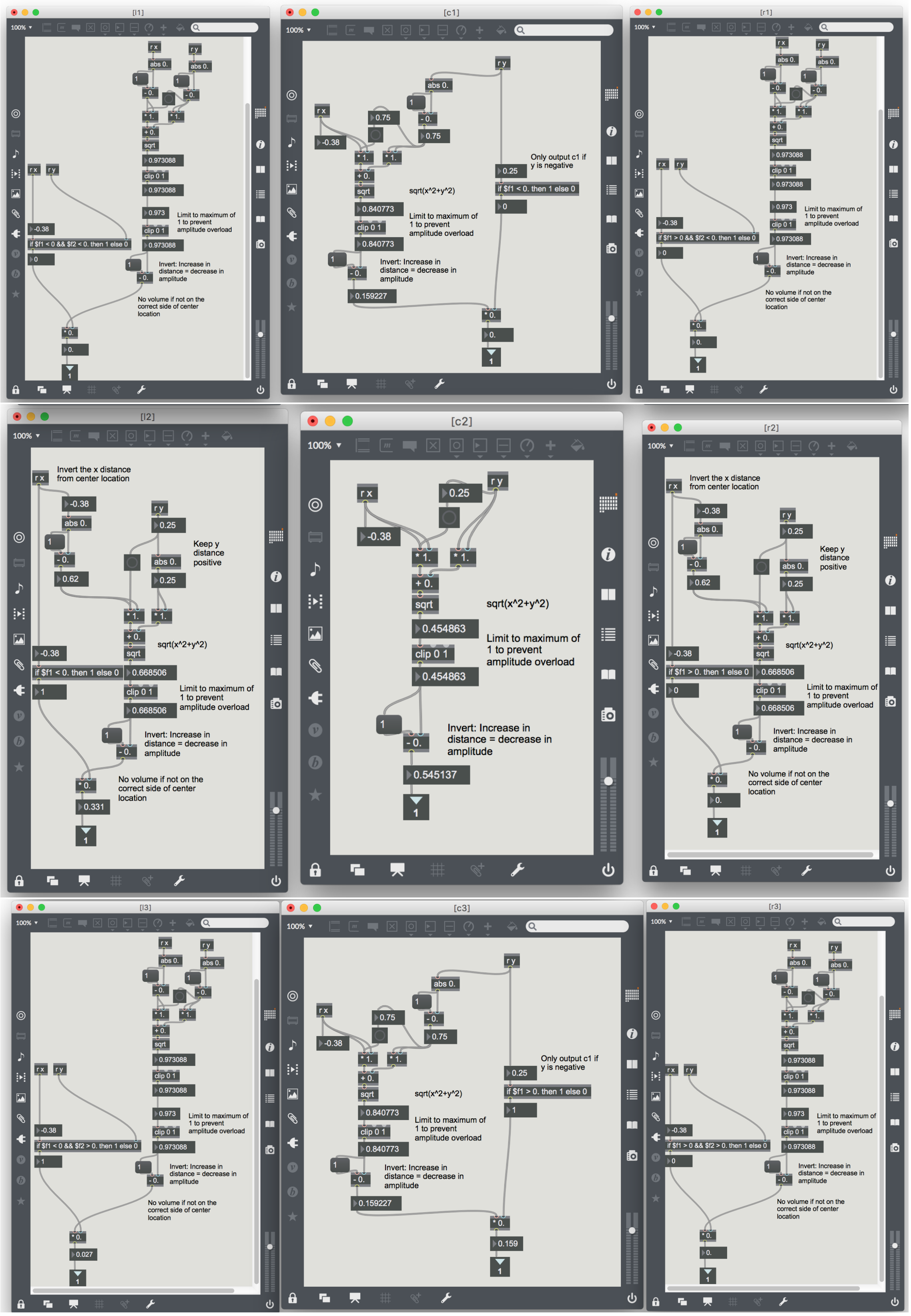


Figure 76: Overview of the individual panning algorithms required for the linear interpolation system used in iteration 1.

Appendix D

Test Participant Form

You have volunteered to partake in two user tests that should take no longer than 30 minutes to complete.

Test Descriptions

The VSS (virtual singing studio) is a system that is used to simulate the acoustics of another room. The system can be used by standing in the centre of the speaker array and singing into a head mounted microphone. By wearing the provided head-tracking device, you can turn in the virtual space by turning your head/body.

Test #1

This test aims to investigate the perception of movement within the virtual acoustic environment when using two different methods: Method **A** and Method **B**. You will be asked to step inside the VSS and say the word "Bob". Your location within the virtual space will then be changed and you will be asked to produce another sound. This process will then be repeated a second time but this time using method **B**. You will then be asked to state whether method **B** felt like you had:

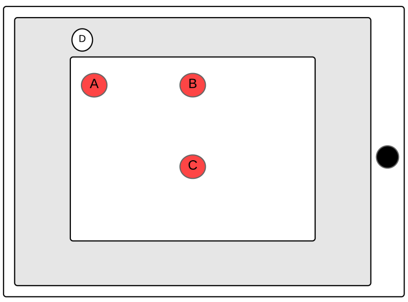
Moved a **shorter** distance than I had in **A**
Moved the **same** distance as I had in **A**
Moved a **further** than I had in **A**
I don't know

This process will be repeated 5 times in total.

Test #2

Part 1: You will be asked to step inside the VSS and to sing or produce a noise. After a short amount of time you will be asked to do the same again. You will then be asked whether you feel you have changed location or not with a simple Yes/No answer. This will be repeated 8 times.

Part 2: In this part of the test, you will be asked to change your location within the virtual space yourself by tapping on a location or dragging your finger around the iPad provided for you. You will be asked to rate on a scale of **1 - 10** how free you feel you can move about the room with **1** being a *jumpy movement* and **10** being *complete freedom to move without limitations*. You will also be given the opportunity to add comments to further explain you score if you wish.



To the left is a diagram of an iPad. **A**, **B**, and **C** indicate where parts of the room can be located. When situated in the VSS, you will start in the center of the room **(C)** facing towards the front of the room **(B)**.

A = Top left corner of the room
B = Front of the room
C = Centre of room
D = Button to calibrate head tracker

Answering Question

Note that when you're within the VSS it will be difficult to write down your answers to the questions asked. Therefore you will be asked to answer verbally and your answers will be taken down for you. You will be asked at the end of the test to check that your answers have been taken down truthfully.

Information and Consent

Experimenter: _______________

Please read the following statements and tick the boxes on the right hand side to indicate that you understand and agree.

I understand that at any point I may choose to withdraw from the experiment	☐
I understand that I may omit answers to any questions	☐
I agree that I am here voluntarily	☐
I understand and agree that the experimenter conductor will be observing the experiment	☐
I agree that the system being used has been explained to me	☐
I agree that the point of this experiment has been explained to me	☐

Participant Signature: _______________

Answer Sheet

Participant Number: _______________ Date: _______________

Test #1

Question 1: Please state whether you feel you have:

Moved a **shorter** distance than I had in **A**
Moved the **same** distance I had in **A**
Moved a **further** distance than I had in **A**
I don't know

Trial	**Score**			
	Shorter	Same	Further	Don't Know
1	○	○	○	○
2	○	○	○	○
3	○	○	○	○
4	○	○	○	○
5	○	○	○	○

I agree that the answers that have been taken down on my behalf are correct □

Participant Signature: _______________

Participant Number: _______________ Date: _______________

Test #2 - Part 1

Question 2: Do you feel you have changed location within the room?

Trial	Answer
1	YES/NO
2	YES/NO
3	YES/NO
4	YES/NO
5	YES/NO
6	YES/NO
7	YES/NO

Test #2 - Part 2

Question 3: Please rate on a scale of **1 - 10** the mobility within the virtual space where **1** = Extremely "jumpy" movement and **10** = Completely smooth movement or please select "N/A" if you can not tell you are moving.

Trial	Score										N/A
1	1 ○	2 ○	3 ○	4 ○	5 ○	6 ○	7 ○	8 ○	9 ○	10 ○	○
2	1 ○	2 ○	3 ○	4 ○	5 ○	6 ○	7 ○	8 ○	9 ○	10 ○	○
3	1 ○	2 ○	3 ○	4 ○	5 ○	6 ○	7 ○	8 ○	9 ○	10 ○	○
4	1 ○	2 ○	3 ○	4 ○	5 ○	6 ○	7 ○	8 ○	9 ○	10 ○	○
5	1 ○	2 ○	3 ○	4 ○	5 ○	6 ○	7 ○	8 ○	9 ○	10 ○	○

Comments:

I agree that the answers that have been taken down on my behalf are correct ☐

Participant Signature: _______________

Question 4: Please rate on a scale of **1 - 10** the mobility within the virtual space where **1** = Extremely staggered movement and **10** = Completely smooth movement or please select "N/A" if you can not tell you are moving.

Trial	**Score**										**N/A**
1	1 ○	2 ○	3 ○	4 ○	5 ○	6 ○	7 ○	8 ○	9 ○	10 ○	○
2	1 ○	2 ○	3 ○	4 ○	5 ○	6 ○	7 ○	8 ○	9 ○	10 ○	○
3	1 ○	2 ○	3 ○	4 ○	5 ○	6 ○	7 ○	8 ○	9 ○	10 ○	○
4	1 ○	2 ○	3 ○	4 ○	5 ○	6 ○	7 ○	8 ○	9 ○	10 ○	○
5	1 ○	2 ○	3 ○	4 ○	5 ○	6 ○	7 ○	8 ○	9 ○	10 ○	○

Comments:

I agree that the answers that have been taken down on my behalf are correct □

Participant Signature: _______________

Appendix E

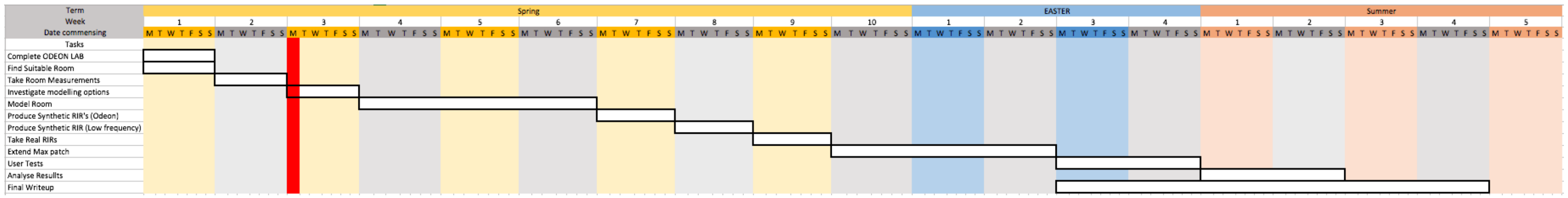


Figure 77: Initial Gantt chart showing the approximate time allocated to each of the tasks required to complete the project. The red line shows the time at which the Gantt chart was abandoned for a more appropriate planning method