School of Electronic Engineering and Computer Science

[Your Degree Here]

Project Report 2017

[Your Thesis Title Here]

[Your Name Here]

Queen Mary University of London

August 2017

1

Acknowledgements

sentimental stuff!!

I also acknowledge that the chapter headings are messed up at the moment, the title page doesn't quite work, the bibtex citation things are broken, the margins are a little large, and I'm sure there's some redundant information up in here.

Contents

2

Abstract

With the recent increase in interest in technologies attempting to provide a virtual experience that convincingly replaces or seamlessly integrates with the real world, it is becoming increasingly important to be able to provide an audio experience that is equally accurate. A common way of reproducing audio for such applications is through the use of HRTFs or HRIRs, models that capture and describe the various effects that human anthropometry have on an audio signal before it arrives at the inner ear. HRTFs, however, are costly to measure, and is has become apparent that the common practice of using an average or generalised HRTF for every user is insufficient, and can cause significant confusion. This project investigates

different HRTF individualisation methods, and attempts to devise and implement a method that would allow users to perform the individualisation process with no additional equipment or expertise.

3

Introduction

3.1 Present Context

Spatial audio has never been more important. Though there has been a steady stream of interest in applications of spatial audio in fields like defence - primarily applied to Virtual Auditory Displays (VADs)(**?**) - virtual, augmented, and mixed reality form a large component of the current technological zeitgeist. One major stated goal of these technologies is that of immersion. This doesn't have to mean that the user feels as if they have been transported to somewhere completely new, they just have to believe in the virtual elements of the experience. To illustrate this, anecdotes about

users gingerly walking around virtual holes, or skirting an object that they know was not in the room before they put the headset on, abound. No matter whether the technology is designed for entertainment, enterprise use, or to assist, the end user must be deceived into believing in what they are experiencing.

For virtual reality to be convincing, audio must have parity with visuals - just small errors in either can irreparably break immersion (**?**). Listeners often complain that auditory events are spatially diffuse, and listeners often make incorrect judgements regarding the source locations (**?**) (Wenzel et al., 1993; Mller et al., 1996?.)

In the real world, humans learn to localise sound sources based on a number of cues naturally encoded in the audio signals arriving at their inner ear. The simplest example of which, inter-aural level difference, or ILD, merely refers to the difference in volume between the listener's left and right ears. Cues like these rely on the fact that humans have two ears, that they are binaural, and so the most effective implementations of spatial audio for virtual reality and other similar technologies attempt to mimic these cues, by filtering audio signals that are mixed down into a two channel - binaural - audio feed, intended for consumption through

headphones (**?**).

Reproducing spatial audio convincingly involves a number of factors, including reflections and occlusion caused by the room and the objects in it. This project however, will focus entirely on the effect the anthropometry of the listener has on the audio signal - the attenuation of the audio signal caused by the various body parts that they sound waves come into contact with, as well as the inter-aural differential cues such as ITD and ILD.

3.2 Modeling Sound Localisation

In most applications involving spatial audio, representing this attenuation is done using Head-Related Transfer Functions, or HRTFs, derived from their time-domain counterparts Head-Related Impulse Responses, or HRIRs. HRTFs are a model for representing this effect on a given signal that the listener's morphology has, and this model can, in theory, be used to convincingly render audio spatially (**?**). HRIR measurements are taken by placing microphones in the ears of a participant (human or mannequin) and measuring the impulse response resulting when a tone is played from a loudspeaker (**?**). This measurement process should be repeated for

as many positions/points of origin around the participant as desired for a given implementation, but can involve as many as 1550 source positions in the case of the ARI(**?**) and SADIE(**?**) databases.This process is incredibly labour-intensive, requires specialist equipment, and can take hours to perform. As a result, there are few organisations capable of performing these measurements, and generating a set of HRTFs for most individuals is impractical at best. There are a few organisations that have assembled databases of HRTFs or HRIRs that involve measurements from a range of participants. Typically, the two main differences in these databases are the number of source positions, and the number of participants involved.

IS THIS SUBSECTION NECESSARY?

3.2.1 Source Positions:

The number of source positions varies from database to database [in the case of CIPIC it is every L degrees from N to M, in the case of ARI, it is every Y degree from X to Z, etc] [add diagrams!]

3.2.2 Subjects:

These databases may contain anything from data from a single mannequin in the case of the MIT KEMAR set (**?**), to the CIPIC database's 45 subjects (**?**), up to the 110-subjects-and-growing ARI HRTF database (**?**).

3.2.3

[table of databases by subjects and participants maybe?]

3.3 The Problem

Because of the aforementioned difficulty in measuring HRTFs, data from these databases is commonly used in attempts to implement spatial audio solutions. Either a participant from the database who is deemed to be sufficiently average in their morphology, a selection of participants, or an HRTF set derived from average values for the participants in the database may be used. In the simplest implementations, there are others but they will go unmentioned for the purposes of this research, the audio sample is then convolved with the HRIR, producing audio that appears to come, convincingly or otherwise, from the position in 3D space that the HRIR was

originally measured from. The difficult task in this case is then to interpolate between these HRIRs in real time in response to the movements of the user (and potentially the source too).

The problem with using this data in any spatial audio implementations that are to be used in applications for the consumption of a wide range of end users, is that HRTF data is incredibly specific to the person the measurements have been taken from. Just small differences in the anthropometry of the measured participant and the end user can compromise the efficacy of the HRTF used(**?**). However, when one tries instead to use a generalised HRTF - derived from the average of a set of measurements, or from a mannequin like the KEMAR(**?**) - the processed audio is ineffective in much the same way that it is when using HRTFs measured from another person. The KEMAR, by dint of being a mannequin with average features, will not be effective for anyone who is not in possession of a totally average morophology. When using HRTFs that are not well matched to the user, front/back and elevation confusion in particular is very pronounced (**?**).

It follows, then, that in a system that implements HRTF-based binaural audio, the audio for a user would be processed using a set of HRTFs that matched the user well enough that the resulting audio would enable the

user to accurately localise the source of a sound. As we have already established, the traditional method of measuring HRTFs is impractical for the vast majority of users, which leaves us at something of an impasse. We need a method for producing individualised sets of HRTFs with minimal specialist equipment, an easy user experience that does not require expert knowledge, as small a time investment as possible.

3.4 Proposed Solution

The method that I am proposing would involve modifying an existing HRTF set so as to better fit a particular user, based on data that can be generated by the user, within a virtual or mixed reality environment. This method assumes the user has access to a virtual reality headset/head mounted display of some kind, and the user will be asked to attempt to locate the invisible source of audio cues that are played to them, within a virtual 3D environment. The data this generates, the difference between the perceived source of the sound and the actual sound source, is what adjustments to the HRTF should be made based upon. This process may then continue until the user starts to successfully localise the sounds sources, or until

the error rate drops below a certain boundary.

This frames the task of HRTF individualisation as an optimisation problem. The goal of a process like the one outlined above being to make certain values, representing the difference between perceived and actual sources, as small as possible. This is of course the implicit goal of every HRTF individualisation method, but including it as a variable in the process allows us to consider adapting existing optimisation search solutions[wording?] for use in this context. During the next chapter I will investigate some existing algorithms, and attempt to gauge their efficacy. The next chapter will also investigate some of the existing models for understanding HRTFs and methods for attaining individualised sets, and whether or not they are applicable to the proposed method.

4

Literature Review

This idea of generating individualised sets of HRTFs without having to perform the complex measurements that would usually be required has existed since the 1990s (**?**). The ideal scenario for commonplace spatial audio involves every user having access to an HRTF set that works for them. If traditional methods of measurements are impractical, then alternatives are necessary.

4.1 Methods

Investigations into HRTF individualisation have been done using a range of methodologies, some involving just simple selection tasks (**?**) and others complex tuning (**?**) - adjusting multiple parameters against listening tests. Often these methods hinge on a specific model that is used to decompose the HRTF into individual parameters that can be manipulated independently in order to achieve meaningful control over the customisation process. In some cases these models also seek to make clear the relationship between the features of the HRTF and the features of the user - the morphological properties of the measuree being the primary determinant of generated HRIR this seems a logical approach. In these next few sections I will cover the main of the approaches that have been investigated to date, as well as their efficacy and why they are or are not well suited to this project. We will see that there is a definite overlap between these methods, leading to the idea that perhaps in a more comprehensive but laborious model for HRTF individualisation, a combination of these techniques might be used (**?**).

4.1.1 Database Matching

Database matching is often incorporated into other models for HRTF individualisation. It is based on the predicate that within a database of a given size, there must be a set of HRTF measurements that have been taken from a participant with similar anthropometric features as a given user. This technique has been used in a range of studies on spatial audio, both as part of a wider study on binaural audio and localisation, (**?**) and as the sole focus of the study (**?**). As in both of these papers from Zotkin et al, many attempts to match participants with closely matching HRTF sets use measurements of the user's anthropometry, which they will then try to match to the anthropometric measurements taken in the process of assembling the database.

This can work reasonably well, assuming the database used contains measurements from a great enough range of people. The CIPIC database contains anthropometric measurements for all 45 of its participants (**?**), while the ARI database comes with measurements for 50 of its participants (**?**). Problems with this method can of course arise when the database does not include measurements from a participant with a morphology that does not closely match those of the user. The second problem with this

method is more of an issue when considering this method in terms of what this project hopes to achieve. Given the my requirements for the customisation method I intend to design, a method that requires precise measurements that would be difficult to perform at home is not ideal. A method that requires precise measurements to be taken has two problems, both in the difficulty of performing the measurements, and the effort that such an act involves, does not satisfy any of my self-imposed standards for user experience.

An alternative method for matching users to their closest-matching HRTF set could be based on subjective listening tests. Playing a user a sample, filtered using an HRTF taken from a database, and asking them to indicate where they believed the sound came from. This process can be repeated for as many examples as are contained in the database, and the one that results in the least incorrect localisation attempts chosen. The problems with this method are again clear, in that the labour required to search all the entries in a database is more than anyone but the most die-hard users are likely to pursue [citation for some human-computer interaction junk about how much effort people will put in?]. Improvements are made on these kinds of subjective listening tests, however, in attempts to match users to

more appropriate HRTFs through the clustering of similar sets.

4.1.2 Clustering

Clustering involves collating a database of HRTF sets measured from different participants, and then sorting these into orthogonal groups based on a specific feature. Fahn and Lo (**?**) grouped HRTFs based on the power cepstra of each HRTF set. They then used a modified version of the LBG algorithm to form 6 different clusters. Other studies, such as Xie et al (**?**) found a total of 7 clusters were required. Either way, the idea is to group HRTFs into groups - or clusters - where each HRTF is similar enough to the others in the cluster, but where the differences between each cluster are sufficiently great. The central example can then be taken from each cluster, the HRTF that best represents that cluster or that represents the average, and provide to the end user the example from this set of 6 or 7 that best matches them.

[must state more clearly how successful these approaches are]

Given that clustering is meant to make simpler the process of matching a user with a more personal HRTF set, trying to match users by anthropometry again would be nonsensical. Instead, subjective listening tests

are used more often (**?**) (**?**). Using this method, the comparative efficacy of subejctive tests in this instance is clear versus raw database matching. As opposed to subjecting an undending barrage of tests against 45 or more (as a slightly facetious example), the listener has to compare between 6 or 7. However, the resulting localisation is going to be less precise, given the inherently more generalised approach. The increase in user-friendliness it interesting, though. In lighter-weight applications of VR/AR, perhaps for example on mobile devices, this approach could work. Giving interested users the option to choose between a subset of sufficiently disparate HRTFs, adding a little lightweight customisation. [HOW SUCCESSFUL IS THIS APPROACH????]

(**?**) (**?**)

4.1.3 Frequency Scaling

Another methodology is based upon scaling in frequency entire HRTFs or elements of the HRTF. A method that was investigated early on in attempts to devise individualisation methods, it is one that lost out to cluster/database matching methods in the longer run.

Some notable examples of studies into this technique include one by

Middlebrooks (**?**). In this study, they used Directional Transfer Functions (DTFs) which are processed HRTFs with the source location information isolated (**?**). Initially finding that spectral features from one participant's DTF could be aligned with those of another by scaling. In further investigation participants used DTFs from the other participants, which were then scaled by a range of different factors based on comparisons in the two participants anthropometry - primarily the size of the head, and pinnae. This study then compared the participant's ability to localise sounds convolved with another's DTF against localisation when using the scaled DTFs and found a roughly 50% increase in accuracy with the most effective scale factor.

Another method investigated by Tan et al (**?**), involved building a tool that allowed users to manipulate the scaling of an HRTF themselves. Given that front/back and elevation confusion is most common when using non-individual HRTFs, they opted to provide options to add a bias towards the front/back, as well as another parameter to tweak how elevation was perceived. Their results showed a small improvement over the non-individualised sets, but the results varied between participants and . Given the simplicity of adjusting a mere two parameters this approach could have been

very convenient. But the lack of an impressive improvement in localisation makes it a less tenable solution than some of the others explored, and it is overshadowed by later methods.

4.1.4 Structural Models

Structural models appear to be the most commonly studied models for understanding HRTFs as well as for attempting to synthesise individual sets or customise generalised sets(**?**). Because HRTFs and HRIRs represent the affects on the sound signal/wave of the features of a human's body, then one should be able to extract and isolate the discrete elements an HRTF that relate to the individual body parts. Klaus Genuit first proposed a model for understanding HRTFs as a series of filters that each represented the effects of a certain anatomical feature (**?**). The idea of a structural model, or of HRTF individualisation based on a user's anthropometry is pervasive, and many other methods incorporate elements from it. For example the aforementioned 2003 study by Zotkin, Duraiswami, Davis, and Hwang (**?**) used anthropometric measurements to match a user to closely-matching set of HRTFs from the CIPIC database. Similarly, later studies centered on Principal Components Analysis - discussed further in

the next section - look at the relationship between principal components (PCs) and morphological features.

Work by Brown and Duda (**?**) (itself based on a 1996 paper (**?**)) looked primarily at HRIRs, focusing on the additional temporal information that the frequency-domain HRTFs lacked. The decision to focus on the time domain was to allow them to identify the characteristics of HRTFs that are the result of the different paths to the inner ear that the sound waves took, over time. This study involved only a small number of participants, and so whether or not the synthesised HRTFs produced with this model could replace measured ones was left to more comprehensive studies.

In a 2001 study Algazi, Duda, Morrison, and Thompson attempted to produce an approximated HRTF from the isolated responses of different structural components (**?**). As with other studies, the synthesis was performed based on anthropometric measurements of the subjects, and the final HRTF composite - made up of the responses of each structural component. This approach was evaluated using a composite HRTF vs a measured HRTF, and when viewed spectrally the two had significant similarities. The study did not go as far as to perform subjective/psychoacoustic tests, however. A similar study in 2003 by Raykar and Duraiswami (**?**)

aimed to decompose the HRTF into a set of significant features that are integral to the localisation of sound sources. Their results were promising, developing an algorithm to decompose a given HRTF, and testing it successfully on every participant in the CIPIC database.

This model often provides promising results, and can be well-suited for applications where a high level of localisation accuracy is required, but a full measurement session is out of the question for the proposed implementation. The main problem with this method being the precision that is required for the measurements. If an ideal implementation for widespread consumer use relies on a simple calibration process, detailed measurements it becomes more difficult to fulfil that requirement. Investigations have been made into the use of computer vision to automate the measurement process and eliminate human error(**?**)[find citation again], but unless this process can be distilled into something simple that requires minimal additional hardware, for example using just a smartphone camera, then it is sub-optimal for widespread use.

More recent studies have tried to combine this approach with others, using a combination of PCA and a reduced number of anthropometric measurements (**?**). Or a similarly reduced number of measurements to

match a set of HRTFs to a subject, then modify them to improve their performance (**?**). Even trying a combination of structural data and a radial basis function (RBF) nerual network (**?**).

4.1.5 Principal Components Analysis

Principal Components Analysis (**?**) (PCA) is a process for analysing a dataset and and identifying the principal components of that dataset. It is a statistical procedure that attempts to return an efficient representation of the dataset, turning the dataset from some large number of variables, into a smaller collection of principal components (PCs), adding some more efficient structure to the dataset.

4.1.5.1 Understanding PCA

PCA can be used to reduce the number of dimensions in a dataset, reducing it down to its most basic components. The dimensionality of a dataset is identified by the number of variables present in that dataset. In general terms, for our current example (that of an HRTF set) the data should represented as a matrix - the structure of the matrix is dependent on the approach being taken, and different approaches will be explored later in this

section. As an example, in the 1992 paper by Kistler and Wightman (**?**) the data set used was arranged into a 5300x150 matrix. If one were performing PCA manually, this dataset could be then be decomposed into a collection of pairs of eigenvectors and eigenvalues - each pair represented by a line through the dataset at the point of greatest variance. Each pair is comprised of a direction and the variance of the data along that line - the vector and the value, respectively. The number of eigenvector/value pairs in a dataset directly corresponds to the dimensionality of the dataset. The dimensions (eigenvectors) with the greatest variance (highest eigenvalues) will consistute the principle components. The number of principle components chosen from the resulting set is dependent on the level of detail necessary. In the aforementioned Kistler and Wightman study, it was found that 90% of the HRTF could be reconstructed using only the five PCs with the highest level of variance. The Principal Component Weights, or PCWs, are the individual values around an eigenvector - the variance between which gives us the eigenvalue.

4.1.5.2 Individualising HRTFs/HRIR with PCA

There are a few main differences among studies that apply principal component analysis to HRTFs and HRIRs. Chief among them is whether the study uses HRTFs (**?**) (**?**) or HRIRs (**?**) (**?**) (**?**). There are benefits of working solely with HRIRs, one retains the interaural time difference (ITD), and it's easier to extract the effects of subject anthropometry on the resultant HRIR. However when analysing HRIRs with PCA, researchers often time-align the HRIRs before processing (**?**), losing information on ITD. Estimation of ITD is a comparatively trivial task, and when not relying on an anthropometric model for individualisation the correlation between HRIR and anthropometric features becomes irrelevant. One advantage of the use of HRTFs that cannot be overstated is the larger corpus of research already done on the work (**?**).

In an early paper on the subject, Middlebrooks (**?**) found that no matter the database used, the amount of variance described by each PC, and the number of PCs required for a substantial reconstruction of the original HRTF was more or less equal. This is helpful, allowing the results of any investigation into a particular method to be applied semi-interchangably to new research. The number of PCs used for reconstruction varies a lot,

however - anywhere from 4 (**?**) to 90 (**?**). The decision as to how many PCs to use depends heavily on the intended accuracy of the reconstruction. The more PCs that are used, the more accurate any reconstruction based on them would be. There are, however, heavily diminishing returns on this number. Many studies are in agreement that using 4-6 PCs can describe around 90% of the variance within the HRTF set (**?**) (**?**). It is common to want to reduce the number of PCs used, so as to in turn reduce the complexity of the tuning process (**?**). When this tuning process is manual, this concern/focus is understandable. But if the process is automated, there could perhaps be a greater focus on accurate reconstruction.

Holzl (**?**) investigated the effect that different input matrices, created by restructuring the HRTF input data, had when performing PCA on HRTFs. In doing this he identified five different matrix arrangements used by different studies, and found the most effective to be [(*subjects * sound directions*) x *signal*] - a structure also used by Kistler and Wightman in their studies (**?**).

There is also a difference in how much of the dataset is analysed and adjusted at a time. Some studies (**?**) [cite a bunch] elect to just adjust a clustered sub-set of positions, playing a subject a sample from that di-

rection and adjusting just those positions based on that. This process of performing PCA and updating the weights for each position can be time consuming. Other studies (**?**) use a more global model, analysing all directions at once. For this method to be practical when performing manual adjustments, a method for modeling PCWs can be helpful. Holzl (**?**) for example proposed a method for modelling PCWs based on the spherical harmonics transform. The proposed model effectively maps the PCWs to a sphere around the subject, allowing manipulation of the PCWs that apply to the intended apparent source of the sound being played to the user at a given time. There is a possibility that this approach could allow automated individualisation to be performed more effectively, adjusting the relevant datapoints in a more targeted way.

4.2 Search Methods

My proposed method for producing HRTFs is based upon being able to automate the adjustment of the principal component weights produced by using PCA on a generic set. The problem with developing a method for individualisation that works in this way is that there is little in the way of

research looking at correlations between localisation errors and principal component weights - which is not altogether surprising. Because of this, I have instead chosen to investigate existing search algorithms that I might be able to fit to this problem. There is a possibility that the data generated by this project could be used to investigate such a correlation, and as such later updates to the process may be able to increase its efficacy. Because of the iterative and interactive nature of this method, any search will be inevitably slow. This is in part because there is no way yet of modelling or forecasting a user's response to a given HRTF-filtered audio sample. Any search method I use must be simple enough to implement easily, but also have the potential to produce a measurable change in localisation performance. Given the potential for incrementally adjusting every HRTF source position to result in a slow individualisation process, an algorithm that helps to alleviate that in any way will be preferable. I have no doubt that with more data regarding error rates and HRTF principal component weights a more sophisticated algorithm could be applied to this problem in the future, but for the purposes of implementing a functioning proof-of-concept a simple approach should be taken.

4.2.1 Hill-Climbing Search

The hill-climbing family of search methods (**?**) start from a given potential solution to a problem, and attempt to find an optimal solution. For our case the steps are so:

- Take a starting state - our generalised HRTF set.
- This state is evaluated - the listener is played a sample and we test how well they can locate the source.
- A change is then made - a predetermined value is added to or subtracted from the parts of the HRTF that corresponds to the direction of the sound source being tested.
- This state is then evaluated again, and the process repeats.

Hill-climbing is notorious for getting stuck in local maxima (**?**), for example if we were to reach a point at which the user was failing to locate a sound source on their localisation attempts, but changes in either direction resulted in even greater error, a hill climbing algorithm would likely move between those points forever. There are ways to work around this limitation, including stochastic hill climbing (**?**) and random restart (**?**), a derivative of that.

For my proposed implementation, hill climbing searches fit the simplicity requirements. Their suitability is more questionable when it comes to the rate of change. Child states in hill climbing search methods are usally generated by making a set change to the current state, deciding how to make these adjustments so that they were both small enough to generate a suitably precise degree of individualisation without it taking an untenably long time to get there.

4.2.2 Genetic Algorithms

realised i mentioned genetic algorithms, so I should probably talk about them

4.2.3 Simulated Annealing

Simulated Annealing (**?**) search is a simple iterative search method that requires a heuristic that measures how close a given state is to the goal state. The steps the algorithm takes are essentially as follows:

- From a starting state - a generalised HRTF.
- The state is then evaluated by some evaluation function - in this case

how close the user got to localising the source a given sample.

- A new pseudorandom state is generated - in this case each PCW would be modified by a random amount, with the degree of randomness depending on the following:
  - If the state was close to the goal state (if the user almost correctly identified the source of the samples) then use less randomness when generating successive states - use a smaller boundary when generating random amounts to adjust by.
  - Otherwise, generate a new state and start from there - allow greater variance in the values that are used to adjust the weights.
- This successive state - a modified HRTF set - should again be tested according to the evaluation function, as the process loops.

This method is of course sub-optimal, because of its loose correlation between the error rates and the adjustments made to the PCs/PCWs. The fact that adjustments made to the HRTF are greater when the localisation error is higher, however, does have the potential to expedite the process. An implementation using this algorithm should produce an improved HRTF set for the user comparatively quickly, after a fewer number of itera-

tions than something like hill climbing search. As such, it has the potential to make the proposed process more user-friendly, in that it is less laborious, while still theoretically having the ability to make small incremental improvements to a near-individualised HRTF set. For both these reasons and the fact that simulated annealing is a relatively simple algorithm to implement, my proof-of-concept implementation will be based primarily on SA.

Some adjustments will need to be made, however, in order to increase the efficiency of the algorithm. If information about changes made to parent states and the error rates they produce is saved, it may help to avoid the problem of pursuing modifications that don't actually get closer to the goal state. This record of how much PCWs are adjusted and the error rates that are produced by those adjustments may help to add a bias to subsequent adjustments made to PCWs. This could limit the amount of randomness required in the adjustments, and may help us to reach the goal state of an individualised HRTF set faster.

5

Methodology

Based on the research outlined in the previous chapter, it was decided that an initial implementation of the proposed process would involve modelling the HRTF using principal components analysis to limit the number of variables in play to those those most significant components. Modifications to this reduced dataset would then be made based on simulated annealing search - with adjustments made where necessary. The efficacy of this implementation of this proposed approach will then be evaluated through listening tests conducted within a simple virtual reality environment. The metric for success is the same as the heuristic being used in the individualisation process - does the participant's ability to localise sound sources

get better over time? If so, by how much?

5.1 Process Notes[what title??]

For this implementation I elected to follow the model for PCA and PCW weight adjustment outlined by Josef Holzl(**?**). The core implementation would largely follow his formulation, as mapped out in the paper, with modifications where necessary. Having a method like this that allows for adaptation of the entire HRTF at once and helps to simplify the calculations that need to take place during the individualisation process, a convenience when a lot of the processing is taking place in real time. The input matrix structure that was decided upon during Holzl's investigation, [(Directions x Subjects)(Frequencies x 2)], was modified slightly to work for a single user to become: [Directions x (Frequencies x 2)]. In practical terms, an entry from the CIPIC HRIR database would be transformed into the frequency domain, resulting in an HRTF dataset in the form [Left/Right (2) x Azimuths (25) x Elevations (50) x Frequencies (101)]. This is then restructured into the above form, resulting in a structure [(Azimuths, Elevations (1250)) x (Frequencies, Left/Right (202))] in size. This structure is intuitive in terms

of how the original values map to the new one, a quality that carries over even after the structure has been transformed with PCA. Performing PCA on this matrix effectively singles out the frequency bins that contain the most variance over all source directions. The resulting [PCW x PC] matrix is equivalent to [Directions x PCs] where the PCs are the frequency bins of greatest variance and the directions the PCWs. The benefit of this resulting structure is that it becomes very easy to modify the PCWs that relate to the position of a sound source. So for my implementation this means that it is simple to map the degree to which a user is able to locate a sound source to a relevant PCW within each PC. Because of this ability to match sound sources with principal component weights, coupled with the fact that PCW modifications were to be automated rather than performed manually, I chose not to model the resulting PCWs using spherical harmonics as Holzl did, further simplifying the individualisation process.

The PCA model produced by this input matrix allows for around ninety percent of the variance in the to be described by 10 PCs, greatly limiting the number of variables that can be modified. This means that to adjust the perceived source of a sample the algorithm needs to only modify ten PCWs, one for each PCW that corresponds to that source position in each

PC.

Core to simulated annealing (SA) search is that the degree of randomness, or the range of potential child states, is reduced as the current state gets closer to the goal state. In this implementation of the algorithm the value that is used to update the PCW is derived from user's localisation error, so the closer the user is to locating the sound source correctly, the smaller the change that is made to the PCW. It is worth noting that this implementation of this search method is effectively running 1250 individual searches in parallel, in order to find the optimal value for every individual source position so the process is unfortunately slow - this quality will be addressed in testing by limiting the number of possible source positions so that the effect of SA on individual source positions over time may be examined.

As mentioned above, modifying PCWs like this means that generating an individualised HRTF set would take at least 1250 seperate measurements, something that is almost as laborious as the standard measurement procedure. So to expedite the process, and to affect a greater amount of the HRTF with each modification, the PCWs for the eight source positions directly around the source being tested will also be modified by

the same value, halved. This may mean that subsequent modifications partially overwrite previous ones, but it is preferable to expedite the process at least while data on the relationship between PCWs and localisation errors is so limited.

Because there will be a not-insignificant amount of time betweeen each alteration made to a given PCW, the details of each alteration made are stored for future measurements from the same source position. This means that each time the algorithm begins to modify a set of PCWs, it first checks the existing database for previous adjustments. If extant, the previous error value, the modifier value for each PCW, and the change directions (whether the modifier was added to or subtracted from the weight within that PC) are all returned, and can inform this and subsequent changes. The direction of change is informed by whether or not this localisation attempt fared better than the previous one. If it did, then the change is made in the same direction. If it did not, some PCWs are adjusted the other way. In the event that a change is reversed and the next localisation attempt is better than the last, this system will ensure that the next change is made in the same direction(s). Given the aforementioned lack of research regarding the effect of the modification of different PCs on the perceived

source of the sound, a decision had to be made regarding whether or not the update value was to be added to or subtracted from the current value of the PCW. The two options were to update every relevant PCW in the same direction, or to vary it. I decided on the latter, using different update directions in order to try and avoid situations where the algorithm might get caught in a loop, oscillating between two near-optimal states. The change was implemented using a process borrowed from genetic algorithms(**?**), in which elements of a bit string are inverted based on a pseudorandom selection process, coupled with the SA's decreasing degrees of randomness. In this implementation a number of indexes in the update-directions matrix are selected at random, but the exact number that are selected is determined again by the degree of error in the user's localisation attempt. This solution should result in steadily more granular changes as the HRTF set becomes more well-suited to the user.

5.2 Implementation

The project that has been produced to test this proposed method is formed of two three parts: A front-end VR test environment that manages the posi-

tions of the sound sources and processes the audio samples, and passes data to another module for processing. This individualisation module forms the core of the implementation, as it handles the deconstruction, modification, and reconstruction of the HRTF data. This module is then tied to an in-memory key-value datastore that holds the starting HRTFs, intermediate results, and an archive of the changes made so far.

5.2.1 Technologies

The bulk of the implementation is written in Python(**?**), making liberal use of the SciPy(**?**) set of libraries to handle the majority of processing involved. Also in use is the simplejson(**?**) module, used to encode both HRTF data and logs extensively. To handle principal components analysis, the scikit-learn(**?**) PCA class from the decomposition module was employed. Lastly, interfacing with the database was done using the lmdb module(**?**). Symas' Lightning Memory-mapped Database (**?**) was chosen both because it is a simple key-value datastore that doesn't require the kind of strictly defined strucure a relational database might, and because the entire contents of it can be loaded into memory when the program is started, making fetch and store operations comparatively rapid - use-

ful when working in real-time. Finally, the test environment was produced sing the Unity game engine (**?**), with the GoogleVR SDK (**?**) and the Final Wireframe (**?**) shader pack.

5.2.2 Process

Typically the individualisation process would run as follows, beginning with the user or participant in the VR space that is being used to generate the data.

- The user faces a marker signifying the direction that relates to the 0, 0 angular coordinate for the CIPIC database.
- The user is then played an audio cue that has been convolved with the corresponding HRIR from a source position that is generated at random but corresonds to a source position available in the database - in the case of CIPIC, this is anything from -45 degrees and up.
- Next, the user should point a reticle situated in the centre of their screen at where they think the sound originated from and issue some kind of confirmation.

- The perceived and actual sound source positions are then passed to the core module.
- This information is then transformed into angular coordinates that match the way the CIPIC data is arranged, from which the update value is calculated like so:
  - maths stuff
- The current individualisation-in-progress HRTF is fetched from the database, along with a pre-prepared PCA model, and reformed into a [1250 x 202] input matrix.
- Next a matrix containing the mean value of each column in this matrix is subtracted from the input matrix and stored for later.
- The input matrix minus the mean is transformed using the PCA model, to produce the [1250 x 10] [PCWs x PCs] matrix.
- The database is then queried, to see if a modification has been made to this source position before, and the update value is used to adjust the PCWs according to the following conditions:
  - If there is no data about previous adjustments, a set of ten

boolean values are generated using Python's random module(**?**) to represent the adjustment direction for each principal component for that this weight (source position).

  - Otherwise, if the data exists and the difference between the perceived and actual source positions in the most recent localisation attempt is greater than the previous one, a randomly chosen subset of the previously-used set of booleans have their values reversed and each PC is adjusted according to those directions. Otherwise, if the difference is lower, make the adjustment in the same direction.

- Information on the details of the adjustment made are then stored in the database, overwriting any previous data for changes made to that direction.

- Once the PC matrix has been updated, the PCA transform is performed in reverse, and the column mean values are added back in.

- Finally, this matrix is re-structured into the same format as the original HRTF matrix and stored in the database under the same key it was fetched from, archiving the previous iteration.

- This new, modified, HRTF is then used to process the next audio sample played to the user, and the process begins again.

5.2.3 Testing

When it comes to analysing the efficacy of the approach, the tests will largely follow the above process with some small deviations. Because of the limited time frame available with each participant, I will be running the process with the random source positions limited to a smaller subset of 12 key directions as opposed to the 1250 possible source positions in the full dataset. Limiting the source positions like this will allow me to ensure that over the course of each test we are able to iterate over each source position several times, getting a better idea of how the PCWs change over time in relation to the error rate, and how localisation accuracy might change for a given direction over time. This limited set of source positions will be located in front of, behind, and to the left and right of the participant, as well as on the eight corners making up a cube around the participant like so:

[DIAGRAM OF THE ARRANGEMENT]

The position of the sound source will change pseudorandomly, ensur-

ing that the subject is tested from each source position the same number of times and that the same position isn't used twice in a row. For each position, the participant will be asked to face forward while the test sample, a one-second clip of pink noise, processed with the (semi)customised hrtf. The participants will have the option to play the sample again if they wish, and will be given ample time to try to locate the source position as accurately as they can. There will also be an on-screen indicator that displays when the participant is facing forward, allowing them to monitor their own head position.

During the tests, the individualisation module will automatically capture and all necessary for analysis. This includes the source location, perceived location, and resulting error value, as well as all of the primary PCWs both before and after they have been updated and what the direction of each update. All of this data is also timestamped, to make it easier to group data by test/participant.

The main metric that I am interested in analysing after the tests is the error rate over time, in particular whether or not there is any decline in error rates as the test progresses. After that, I will also be looking to investigate whether or not there is any relationship between localisation errors,

individual principal components, and update directions. Any potential relationship that exists there is interesting, because that information could be used to substantially enhance any future implementations of/updates to this individualisation process. There is enough variety in the captured data that answers to other questions may be investigated if time permits.

6

Analysis

I was thinking I'd do a section for each metric/relationship I'm able to analyse

6.1 Localisation Error Over Time

start with description of the data being displayed on the charts and how the charts were generated (how data was processed and the technologies used)

Here I was thinking a graph or set of line charts displaying mean error rate for all participants over time for each source position, but twelve lines

on a single chart might be too much? Unless it's one very large chart. So I was expecting to split it into three, one for each elevation - sources below the listener, sources at the same level, and sources above. There will also be only about 5 points along the X axis so they might all fit in a single line. Is there a well-defined style? Or should I just try both and go with whatever I think looks nicer?

Then describe what the conclusions can be drawn from the data being displayed

6.2 Relationships Between PCs and Localisation Errors

Again, starting with a description of how the data was was collated from the tests, and how it was arranged on the charts. I just can't yet work out the best way of displaying the data for this. It would be nice to represent it on a sphere or section of a sphere, something like that?

same structure, charts in between

Conclusions we can draw, etc

6.3

Then this structure could be repeated as needed, for as much analysis as I get to do??

7

Discussion

- Does the algorithm work? (noting flaws)
  - If no, why not, what could have made it work?
  - If yes, how could it be improved?
- *elisa from hamilton voice* what would I have done if I'd only had time?
- what was out of scope for this project, but would be an interesting extension of this? collating more Error/PC data and using it to train an ANN or other more interesting ML model? how could anything, if anything, be applied to future stuff

Bibliography

Josef Hölzl. *An initial Investigation into HRTF Adaptation using PCA IEM Project Thesis*. PhD thesis, Graz University of Technology, 2012. URL `https://github.com/jhoelzl/HRTF-Individualization`.

8

Appendix