School of Electronic Engineering and Computer Science

MSc Computing and Information Systems

Project Report 2017

# Individualising Head-Related Transfer Functions with principal components analysis and simulated annealing

**Robin Farrow-Yonge**



Queen Mary University of London

August 2017

## 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 Head-Related Transfer Functions (HRTFs) or Head-Related Impulse Responses (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 it has become apparent that the common practice of using an average or generalised HRTF for every user is insufficient and can cause significant localisation confusion. I will be investigating different HRTF individualisation methods and attempting to implement a process based on Principal Components Analysis and Simulated Annealing search; allowing users to perform the individualisation process with no additional equipment or expertise. This method is shown to produce somewhat promising results, but could be significantly improved by further investigation into the relationships between the components singled out by PCA and localisation errors.

**Acknowledgements**

I would like to acknowledge a number of people whose input on this project, whether directly applied or by proxy, helped shape the final outcome. Whether successful or otherwise, this work has taught me so much, and I will forever be indebted to those who helped me complete it.

- My supervisor, Dr Rebecca Stewart, without whose patience in the face of panicked emails and expertise in the subject area I fear I would have scuttled this project very quickly.

- All of my coursemates, for the myriad ways you've all helped over this past year and for dragging me out to socialise on occasion.

- My parents both for listening to me freak out at various points, and letting me cancel on them so I could work more.

- My internet family: Nix, Jake, Pottsy, Roxy, Pancake, Ash, Sam, Forrest and Holden. Thank you for encouraging and supporting me, keeping me sane, and ensuring I had at least some human contact on days when I was at my desk from dawn 'til dusk.

- Lastly, to all the friends that I've neglected - We cool? Cool.

# Contents

# 1

# Introduction

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)(Bronkhorst et al., 1996) - 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 at a minimum have parity with visuals as just small errors in either can irreparably damage immersion. Listeners often complain that auditory events are spatially diffuse, and listeners often make incorrect judgements regarding the source locations (Wenzel et al., 1993).

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. This is achieved by filtering audio signals that are mixed down into a two channel audio feed, intended for consumption through headphones (Algazi and Duda, 2011).

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. That is, the attenuation of the audio signal caused by the various body parts that the sound waves come into contact with, as well as the inter-aural differential cues such as ITD and ILD.

## 1.1 Modeling Sound Localisation

In most applications involving spatial audio, representing this attenuation is done using Head-Related Transfer Functions (HRTFs) derived from their time-domain counterparts Head-Related Impulse Responses (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 (Gardner and Martin, 1995) (Algazi et al., 2001). 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(Field, 2017) and SADIE(Kearney, 2017) 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.

## 1.2 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 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(Middlebrooks, 1999). 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 morphology. When using HRTFs that are not well matched to the user, front/back and elevation confusion in particular is very pronounced (Wenzel et al., 1993).

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.

## 1.3 Proposed Solution

The method that I am proposing is to modify 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 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 for use in this context. In chapter two I will investigate some existing algorithms, and attempt to gauge their efficacy. This chapter also investigates 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. Chapter three details how the proposed solution has been implemented and tested, as well as notable considerations and decisions that had to be made, and the technologies involved. Chapter four covers the analysis of the data from the listening tests, both in order to ascertain whether or not the project can be deemed a success and to identify data that may be used to inform future updates to this work. Lastly, chapter five dicusses these findings in comparison to the stated goals of this research, as well as proposing some key improvements that may be applied to future implementations of similar systems.

# 2

# 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 (Kistler, Doris J; Wightman, 1992). 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.

## 2.1 Methods

Investigations into HRTF individualisation have been done using a range of methodologies, some involving just simple selection tasks (Zotkin et al., 2002b) and others complex tuning (Tan and Gan, 1998) - adjusting multiple parameters against listening tests. Often these methods hinge on a specific model that is used to decompose the HRTF into individual parameters. These 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 participant being the primary determinant of a given HRTF 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. I will demonstrate 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 (Hoene et al., 2017).

### 2.1.1 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, (Zotkin et al., 2002b) and as the sole focus of the study (Zotkin et al., 2002a). 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 (Algazi et al., 2001), while the ARI database comes with measurements for 50 of its participants (Acoustics Research Institute, 2017). 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 my stated requirements, any method that requires precise anthropometric measurements is problematic. This is due to both the difficulty of performing the measurements effectively, and the effort that such an act involves, the use of either of which would not satisfy any of my self-imposed user experience standards.

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. 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.

### 2.1.2 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 (Fahn and Lo, 2003) 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 (Xie et al., 2013) 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 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. This approach appears largely effective, Shimada et al (Shimada et al., 1994) found that their participants exhibited significantly improved localisation performance using their clustering method.

Given that clustering is meant to facilitate 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 (Xie et al., 2013). Using this method, the comparative efficacy of subejctive tests in this instance is clear versus raw database matching. As opposed to subjecting an unending barrage of tests against 45 or more (as a slightly facetious example), the listener has to only 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.

### 2.1.3 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 (Middlebrooks, 1999). In this study, they used Directional Transfer Functions (DTFs) which are processed HRTFs with the source location information isolated (Middlebrooks and Green, 1990). 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 (Tan and Gan, 1998), 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 nonindividual 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; 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.

### 2.1.4 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(Brown and Duda, 1998). 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 (Genuit, 1984). 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 (Brown and Duda, 1998) (itself based on a 1996 paper (Lopez-Poveda and Meddis, 1996)) 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 (Algazi et al., 2001). 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 2004 by Raykar and Duraiswami (Raykar et al., 2004) 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 is 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(Mohan et al., 2003), 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 (**?**).

### 2.1.5 Principal Components Analysis

Principal Components Analysis (Pearson, 1901) (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.

#### 2.1.5.1 Understanding PCA

PCA can be used to decrease 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, Kistler and Wightman (Kistler, Doris J; Wightman, 1992) used a data set 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.

#### 2.1.5.2 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 (Hölzl, 2014) (**?**) or HRIRs (Hwang and Park, 2008) (Hwang and Park, 2007) (Fink and Ray, 2012). 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 (Hwang et al., 2010), 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 (Hölzl, 2014).

In an early paper on the subject, Middlebrooks (Middlebrooks and Green, 1992) 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 (Martens, 1987) to 90 (Martens, 1987). 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 (Martens, 1987) (Kistler, Doris J; Wightman, 1992). It is common to want to reduce the number of PCs used, so as to in turn reduce the complexity of the tuning process (Hwang and Park, 2007). When this tuning process is manual, this concern/focus is understandable. If the process is automated, there could perhaps be a greater focus on accurate reconstruction.

Holzl (Hölzl, 2012) 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 (Kistler, Doris J; Wightman, 1992).

There is also a difference in how much of the dataset is analysed and adjusted at a time. Some studies (Hwang and Park, 2007) elect to just adjust a clustered sub-set of positions, playing a subject a sample from that direction 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 (Hölzl, 2014) 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 (Hölzl, 2014) 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.

## 2.2 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.

### 2.2.1 Hill-Climbing Search

The hill-climbing family of search methods (Ligeza, 1995) 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 (Ligeza, 1995), 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.

### 2.2.2 Genetic Algorithms

Genetic algorithms (GAs) (**?**) are a broad categorisation of a family of algorithms, whose operations mimic evolutionary processes. They have a degree more complexity than the other two algorithms mentioned here, but were quickly ommitted from consideration as something I could base the individualisation process on purely because it would not be practical to modify the steps involved for this use case. The reason they are worth mentioning at all is because the final implementation borrows one of the three primary operations involved in a typical GA. Broadly speaking, GAs work on a pool of candidate states. These states are generated in a pseudorandom way at the beginnig of the process, and evaluated to determine their fitness. A set pool of the fittest candidates are then selected as the next generation, usually referred to as the selection stage. This generation is then subjected to the following operations:

- During the crossover stage, candidates are selected pseudrandomly from the current pool. Then at a random point within each candidate, the candidates are split, and half of one will be switched with half of another to create two new states.

- Then, during the mutation stage, elements of a subset of states may be changed. For example, if a given pool of states are represented as binary strings, the mutation may involve inverting a single bit somewhere in the string. Often the algorithm is steered toward mutating parts of the chosen states that would otherwise not be modified with crossover alone.

Given that the evaluation function in this instance would have to be a user in a listening test, this would not be practical for this problem. I have however borrowed the basic functionality behind the mutation stage to inject some randomness into the makeup matrix that dictates whether the adjustment value is subtracted or added to a given PCW.

### 2.2.3 Simulated Annealing

Simulated Annealing (van Laarhoven and Aarts, 1987) 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 for this individualisation process 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 iterations 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.

Based on the literature reviewed, an individualisation process centered around principal components analysis and simulated annealing search was selected. With the aforementioned user experience requirements, a method that allowed elements of the HRTF to be modified without information on the relationship between localisation errors and principal component weights was important. PCA and search methods like SA complement each other well too. The application of PCA serves to reduce the number values being manipulated in the serach process, and therefore the number of potential child states with each step in the search. Simulated annealing is being chosen in an effort to expedite the early stages of the search process while allowing for more micro changes as error rates lower. In the next chapter I will outline the final implementation being used to test the algorithm, as well as the problem-specific decisions that have been made in the process of building it.

# 3

# Methodology

Based on the research outlined in the previous chapter, I 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 the proposed approach has then been 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?

## 3.1 Algorithmic Design

### 3.1.1 PCA

For this implementation I elected to follow the model for PCA and PCW adjustment outlined by Josef Holzl(Hölzl, 2014). The core implementation would largely follow his formulation, as mapped out in the paper, with modifications where necessary. This method 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*) x (*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.

### 3.1.2 Simulated Annealing

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 is 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.

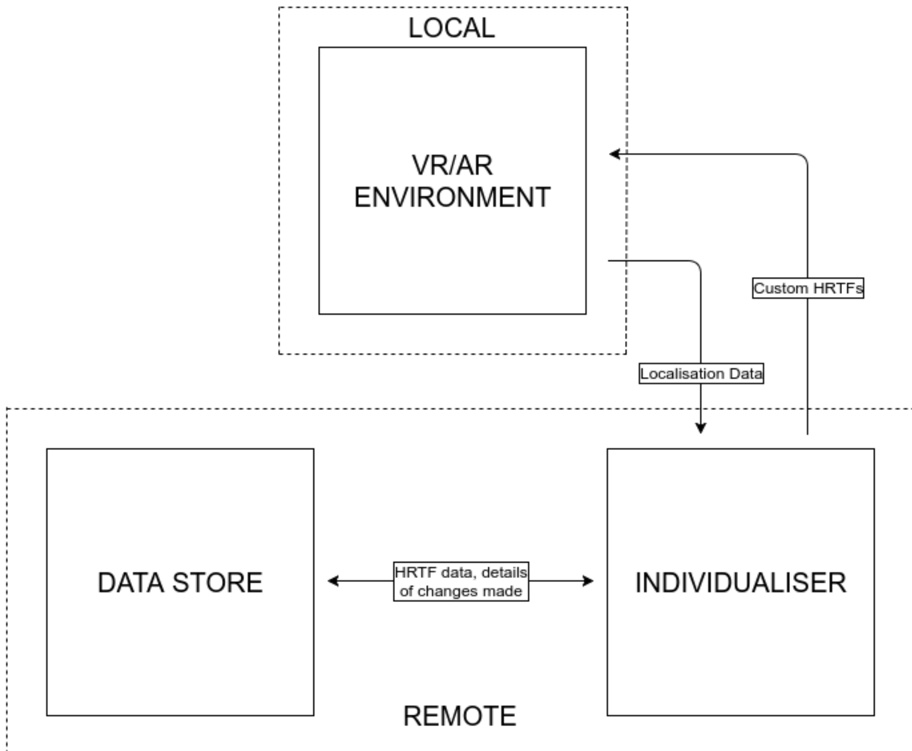
As mentioned above, modifying PCWs like this means that generating an individualised HRTF set would take at least 1250 seperate measurements, something that is almost as laborious as the standard measurement procedure. To expedite the process, and to affect a greater amount of the HRTF with each modification, 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. This feature becomes somewhat irrelevant in the actual listening tests, but is of interest when considering the practical application of the algorithm.

Because there is a significant 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 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.

## 3.2 Implementation

The project that has been produced to test this proposed method is formed of three parts. First, a VR test environment that manages the positions of the sound sources and processes the audio samples, passing data to another module for processing. Next 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.

Figure 3.1: HRTF Individualiser system diagram.




In this test implementation the VR environment and individualisation module run on different machines. This was necessary for ease of development, but it is also an interesting idea for potential future implementations using a similar method. If such a system could be built to aggregate PCW/localisation error data, using it to train a model that takes that data and returns customised HRTFs, it could benefit from the input of a larger amount of data than a method that ran exclusively locally.

### 3.2.1 Technologies

The bulk of the implementation is written in Python (Guido van Rossum, 2010), making liberal use of the SciPy(Jones et al., 2001) set of libraries to handle the majority of processing involved. Also in use is the simplejson(Ippolito, 2017) module, used to encode both HRTF data and logs extensively. To handle principal components analysis, the scikit-learn(Pedregosa et al., 2011) PCA class from the decomposition module was employed. Lastly, interfacing with the database was done using the lmdb wrapper module(David Wilson, 2017). Symas' Lightning Memory-Mapped Database (Chu and Symas, 2011) 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 - useful when working in real-time. The test environment was produced using the Unity game engine (Unity Technologies, 2017), with the GoogleVR SDK (Developers Google, 2016) and the Final Wireframe (Project Wilberforce, 2017) shader pack, and run on a smartphone inside a Daydream(Google, 2016) viewer. Lastly, the individualiser module and production database are hosted on an Amazon Web Services (AWS) EC2 instance and communicate with the VR environment via TCP.

### 3.2.2 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 the CIPIC database, 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:

  - The perceived source position is subtracted from the actual source position, and divided by ten.
  - These two values are each multiplied by a weight value, which was set to 0.6 during the listening tests, and summed together.

  This process is somewhat arbitrary, and mostly serves to generate a value between zero and three. This threshold is based on Holzl's implementation, and introduction of the weight value into the calculation process is primarily to allow easier adjustment prior to testing.

- The current individualisation-in-progress HRTF is then 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 resulting matrix 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.

### 3.2.3 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 available with each participant, I have run the process with the random source positions limited to a smaller subset of 8 key directions as opposed to the 1250 possible source positions if working with the full dataset. Limiting the source positions like this has allowed 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 is 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.

The position of the sound source will change pseudorandomly, ensuring 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 is asked to face forward while the test sample, a one-second clip of pink noise, processed with the in-progress custom HRTF. The participants will have the option to play the sample again if they wish, and are given as much time as they require 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 all the data necessary for analysis. This includes the source location, perceived location, and resulting error value, 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 later extract individual tests.

The main metric that I have analysed based on the tests is the error rate over time, specifically to ascertain whether or not there is any decline in error rates as the test progresses. As a secondary point of interest, I have tried to investigate whether or not any relationship between localisation errors, individual principal components, and update directions can be identified in my data. Any potential relationship that exists there is of note, because that information could be used to substantially enhance any future implementations of/updates to this individualisation process.

# 4

# Evaluation

The planned listening tests were performed on a total of twelve participants, of which ten were usable due to inconsistencies in the testing process, with ages ranging from 21 to 30. In most of the tests, the overall user experience was deemed intuitive. There was some degree of confusion with regard to the features of the UI the participants were presented with, especially the cubes used to mark which way the participant was facing. More than half of the subjects tested at least initially expected the sounds to come from these markers, and some mentioned that sounds continued to appear to originate from them even after they had been made aware. In future tests, UI elements or position indicators should not have such a physical presence. A shortened version of the test was performed to allow the participants to acclimatise to the space, as well as to allow them to clarify any points of uncertainty before beginning the test proper. The tests themselves then took in between thirteen and nineteen minutes, and the primary bottleneck was the time spent preparing the next sample and fetching the modified HRTF data, which was dependent on network speeds at the time. Ideally, any future versions of the test would be more self-contained. If the entirety of the process could be user-controlled, as opposed to requiring regular interaction between the subject and myself, then it may help the user to be better immersed in the process - perhaps producing better results.

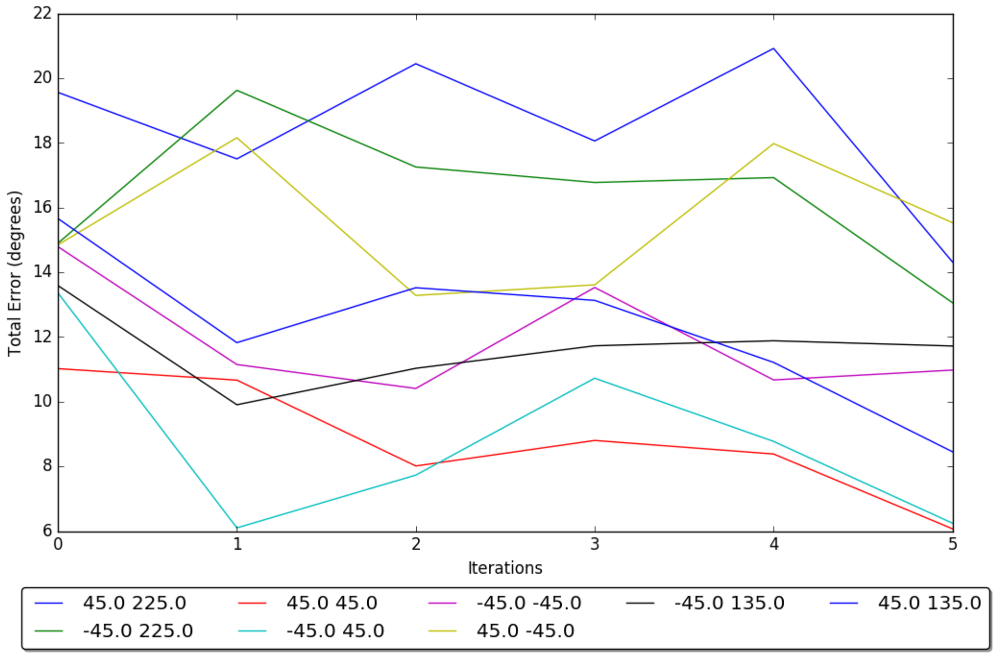
As mentioned in previous chapters, the primary metric for whether or not this implementation can be deemed effective is based upon the degree to which the severity of a participant's localisation errors decrease over time, if they decrease at all. This chapter will begin by investigating whether or not this is the case by reviewing the data captured during the listening tests. After this, an additional section will investigate the aforementioned relationship between PCW adjustment and localisation. To this end I have compared PCW values and perceived/actual source positions before and after different adjustment in an attempt to ascertain whether or not this is something that warrants additional study.

## 4.1 Localisation Error Over Time

In order to analyse this metric, I have looked at the average total error value for each of the eight source positions. This error value is the angular error that comes from subtracting the perceived source position from the actual source position. These two values are then made positive if either one is negative, so as to better represent the total error in degrees, and summed to produce a combined error value. This process is performed for every participant, and the mean error value for each measurement over the course of the test from each direction is then calculated based on that data. These mean error rates over time can then be plotted as in figure 4.1.

On this chart, each line represents a source position, and tracks how, on average, localisation errors for samples played from that source position changed over the course of the listening test. When displayed like this we can see something of a reduction in the overall errors for most source positions, but the short test length makes it unclear how close to zero these values would get or how long it would take. What we can see at play is the self-correcting quality afforded to the algorithm through knowledge of the history of previous changes. For almost all of the directions there are large spikes where the algorithm explores a sub-optimal child state and the participant's subsequent localisation attempt fares more poorly. In all

Figure 4.1: Average error rate over each source position.




cases but one (the black line representing the [-45.0, 135.0] source position) this mistake is quickly rectified as the algorithm makes adjustments in the opposite direction. Based on this and the slight overall trend down, it would not be unreasonable to expect that on a long enough timeline, the error rate would eventually get close to zero. If successive child states could be selected more intelligently, then it's quite possible that these kinds of mis-steps could be minimized further, reaching a more effectively individualised HRTF sooner.

It's also worth noting that three of the four directions with the most promising results were situated in front of the participant: [-45.0, 45.0], [45.0, 45.0], and

[-45.0, -45.0]. This might stem from the much-documented front-back confusion common in non-individualised HRTFs, leading to poor average error rates for sources positioned behind the participant. Future implementations that are better able to account for head movement, allowing the participant small movements to aid in their localisation attempts, might produce better results here. There is an exception to this in position [45.0, 135.0], which may have something to do with the starting HRTF used. Further research would be necessary to identify whether this is truly the case, but it would be interesting to investigate whether or not it is possible to generate a better base HRTF to begin this process with.

## 4.2 Relationships Between PCs and Localisation Errors

The ideal outcome when investigating this relationship would be to find a very clear indication of some correlation between certain combinations of adjustments to subsets of the available PCWs, and a change in apparent sound source. The starting dataset for this analysis drew from all the data captured, which was then sorted into a matrix in the shape [(Subjects x Directions) x Test iterations]. The data of interest from these tests was how the subject's perception of the location of the sound changed - in which direction and by how much - and which PCWs were adjusted to elicit that change.

To this end, the six tests for each direction for each subject were organised into five pair objects, each of which contained the PCWs and the perceived location of the sound before and after. Because the location information in each pair object represented an apparent source position, fixed to the inside of a sphere with the subject in the centre of it, the difference between the two positions describes a perceived change in position. As an attempt at limiting the scope of this analysis, I chose to classify each pair according to whether the sound source appeared to move up-left, up-right, down-left, or down-right. These four classes are sufficient to decsribe every perceived movement, due to the test being limited exclusively to sound sources on the inside of a fixed-size sphere. The degree to which a given sound source moves between the two axes of each of these directional quadrants may vary, and of course more focused research would be necessary in order to gain a more fine understanding.

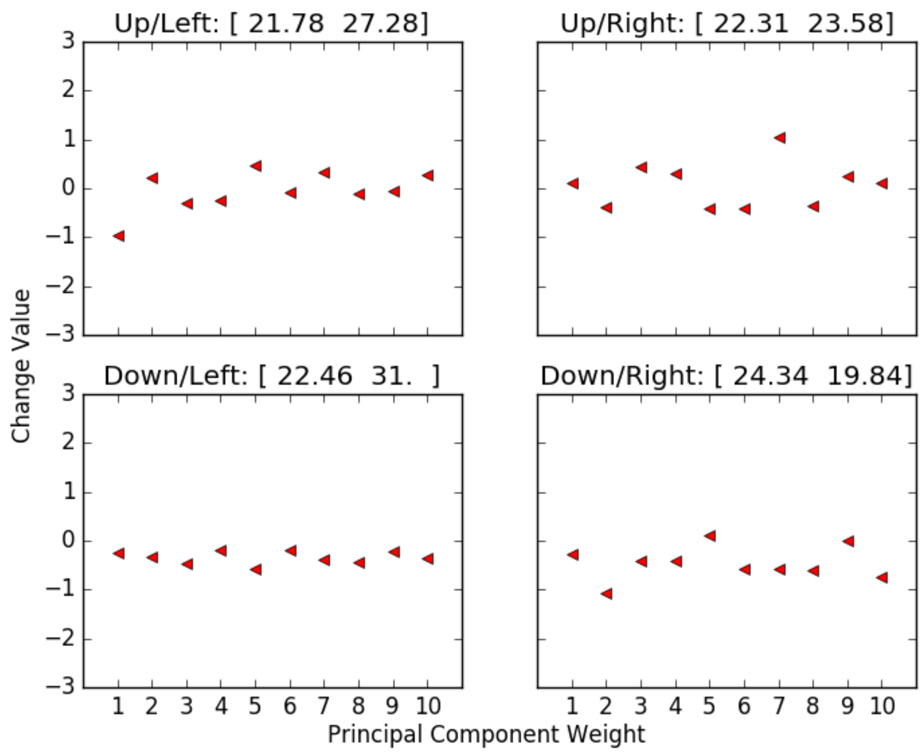
Each of the four lists were then filtered to eliminate any pairs where a sound source appeared to move upwards of ninety degrees between two subsequent tests. Such instances had the potential to be caused by mistakes by the user, or confusion early in the testing process, and it could not confidently be claimed that they represented deliberate localisation attempts.

In order to generate the charts in figure 4.2, the average PCW adjustments were calculated for each change direction, and were then plotted according to how much each PCW was modified by, on average, to generate an apparent positional change in the given direction. In addition to this, the average amount of change for each direction is represented as a pair of values, in degrees, above each plot. The rationale behind this being that the average adjustment should result in something similar to the average perceptual change in position.

There is a slight disconnect in the results as they appear on the chart representing up-right perceptual changes, compared to the others. The average apparent change in that direction is similar to the other three, but the changes made to the PCWs appear much more sweeping. More pair objects were filtered out for this quadrant than any other, from a set that was already the smallest of the four, so it's likely that with more data this graph would be as smooth as the others.

From this small initial chunk of analysis we can see that there does appear to be a distinct combination of PCW adjustments that results in the kind of positional change denoted by their categorisation. However, because the different adjustments use different combinations of the same weights, there will no doubt be overlap when trying to apply this in practice. Adjusting the PCWs for an HRTF

Figure 4.2: Average PCW adjustment for each perceptual position change.



according to the values in the up-left chart might not simply move the source up and left, because some of the changes are shared by multiple directions. For example in this case, the adjustments to the first five PCWs to attempt to move the source position up and left are very similar to the adjustments made to the first five PCWs in a down-right movement. The first PCW, which contains the most information and is most influential in the eventual reconstruction, is adjusted in the opposite direction in each one, so the combinations are distinct overall, but it is not yet known how much of an effect this may have in any resulting perceptual tests. Similarly, in the down-left and up-right adjustments, we see similar values for the first and fourth PCWs, and differences in PCWs two, three, and five. These similarities are the inverse of those on the up-left and down-right charts, which is promising, suggesting a neat correlation might exist when positioning sources towards these quadrants. I have omitted discussion of PCWs six to ten so far, primarily because of their comparatively minimal influence, but it is worth noting that in both the up-left and up-right directions the value of PCW six is increased, though comparatively minimally in the up-left case, possibly indicating some effect on perception. There is a similar link in both the down-left and down-right directions, in which the eighth PCW is increased slightly. Though this adjustment is a lot less pronounced, so further research should be done to confirm any correlation.

# 5

# Discussion

Introducing this project, I defined my goal as being to produce a process with which to generate a individualised HRTF set based on user-generated data, without any more equipment or expertise than the average end user of a virtual, augmented, or mixed reality system might have, on a practical time scale. The process as it currently stands does not effectively meet these requirements, though I am confident that it is a good starting point - serving its function as a proof-of-concept. From a user experience standpoint, basing the individualisation process on the user's error in localisation is an approach which has the potential to be effective. There is sizable scope for gamification of the process using this method, which has the potential to encourage non-enthusiast users to undergo the process as these technologies become more widespread and their application areas become more diverse. To make it practical, however, the underlying algorithm would need to produce an individualised HRTF using a smaller amount of direct input from the user than is currently required. Using the current implementation, even if the rate of measurements were to be increased tenfold it would still take an untenably long time to produce an entire HRTF set, losing a potential point in favour of this type of method over the traditional measurement process.

This leads into the greatest current problem with the produced implementation; it is unknown whether or not it would actually produce an effective custom HRTF set. This is part of the problem of choosing to base this project on SA, as I have discussed in previous chapters. Making pseudorandom changes to the HRTF data means that sub-optimal child states are regularly explored, and this slows the process down significantly. Given that my listening tests were limited to exploring six steps into the search process, it is difficult to estimate how many steps through the search would be required to reach a point at which the HRTF could be said to be fit the individual. Knowing, however, that it is greater than six means that it can be conluded that individualisation of a full set of HRTFs using this exact implementation is untenable.

## 5.1 Improvements

It is possible that some small modifications could make this method more effective, starting with the effects on perception that result from a given adjustment value. The current implementation was based on Josef Holzl's implementation(Hölzl, 2014), that allowed adding or subtracting values up to three from the PCWs, but a more concerted look into the effects of different ways of calculating the adjustment value could be beneficial. It would also be worth reviewing the effect that different starting HRTF sets have on the process. As has been mentioned previously, it's impossible to find HRTF data that works for everyone. But it might be possible to find a small set that work broadly as starting points for individualisation. The starting HRTF could even be selected automatically based on a similar style of localisation test.

It has already been covered heavily in previous chapters, but a better understanding of the relationship between the PCW adjustments being made and the effect that the adjustments will have on perception could be the greatest improvement while keeping the broader implementation details largely the same. We saw in the previous chapter that there appears to be some correlation between PCW adjustment and perceived source position, and with more focused research in this area it's possible that a much clearer relationship could be identified. Coupled with a better understanding of the change produced by a given adjustment value, this could help eliminate the early missteps in the search process and allow the algorithm to make more targeted adjustments sooner, greatly lowering the total number of steps required for individualisation.

Working from the assumption that enough data can be gathered, potentially through use of or research into a greatly improved version of the current algorithm, I think there is also scope to investigate the application of more complex machine learning algorithms to this process. There are plenty of obstacles ahead of making this feasible, but assuming they could all be overcome with sufficient research time then the ideal implementation might function as follows: As with the current implementation, it should start with a neutral/default HRTF, play the user a set of sounds from a predetermined set of positions, and ask them to locate where they believe they originated from. This data should be enough (the exact amount of data quantified with the design of the system) to feed into a pre-trained machine learning model that should then return a more individualised HRTF. Training such a model would take more data than appears to be currently available, and the time required to generate that data could be substantial. Though if there was the potential to build the ability to capture that data into a simpler system that was actively used, it could eventually be feasible. Such a system, fantastical though it may currently be, would fulfil all of the requirements that I initially defined, while hopefully offering improved an improved spatial audio experience for a wider range of people.

# Bibliography

Acoustics Research Institute. ARI HRTF Database, 2017. URL `https://www.kfs.oeaw.ac.at/index.php?view=article{&}id=608{&}lang=en`.

V. Algazi and Richard Duda. Headphone-Based Spatial Sound. *IEEE Signal Process. Mag.*, 28(1):33–42, jan 2011. ISSN 1053-5888. doi: 10.1109/MSP.2010.938756. URL `http://ieeexplore.ieee.org/document/5670436/`.

V Ralph Algazi, Richard O Duda, Dennis M Thompson, and C Avendano. THE CIPIC HRTF DATABASE. In *Proc. 2001 IEEE Work. Appl. Signal Process. to Audio Acoust.*, number October, pages 99–102, New Platz, NY, 2001. doi: 10.1109. URL `http://ieeexplore.ieee.org/stamp/stamp.jsp?tp={&}arnumber=969552{&}isnumber=20913`.

Adelbert W Bronkhorst, J A (Hans) Veltman, and Leo Van Breda. Application of a Three-Dimensional Auditory Display in a Flight Task. *Hum. Factors*, 38

(1):23–33, 1996. doi: 10.1518/001872096778940859. URL http://dx.doi.org/10.1518/001872096778940859.

C.P. Brown and R.O. Duda. A structural model for binaural sound synthesis. *IEEE Trans. Speech Audio Process.*, 6(5):476–488, 1998. ISSN 10636676. doi: 10.1109/89.709673. URL http://ieeexplore.ieee.org/document/709673/.

Howard Chu and Symas. Open LDAP Lightning Memory-mapped Database, 2011. URL http://www.openldap.org/software/repo.html.

David Wilson. lmdb 0.93: Universal {Python} binding for the LMDB 'Lightning' Database. 2017. URL http://github.com/dw/py-lmdb/.

Developers Google. Google VR SDK for Unity — Google VR — Google Developers, 2016. URL https://developers.google.com/vr/unity/.

C. S Fahn and Y. C Lo. On the clustering of head-related transfer functions used for 3-D sound localization. *J. Inf. Sci. Eng.*, 19(1):141–157, 2003. ISSN 1016-2364.

Free Field. Acoustics Research Institute, 2017. URL https://www.kfs.oeaw.

ac.at/index.php?option=com{_}content{&}view=article{&}id=608{&}Itemid=606{&}lang=en{#}tools.

Kimberly J. Fink and Laura Ray. Tuning principal component weights to individualize HRTFs. In *ICASSP, IEEE Int. Conf. Acoust. Speech Signal Process. - Proc.*, pages 389–392, Kyoto, mar 2012. IEEE. ISBN 9781467300469. doi: 10.1109/ICASSP.2012.6287898. URL http://ieeexplore.ieee.org/document/6287898/http://ieeexplore.ieee.org/stamp/stamp.jsp?tp={&}arnumber=6287898{&}isnumber=6287775.

William G. Gardner and Keith D. Martin. HRTF measurements of a KEMAR. *J. Acoust. Soc. Am.*, 97(6):3907–3908, jun 1995. ISSN 0001-4966. doi: 10.1121/1.412407. URL http://asa.scitation.org/doi/10.1121/1.412407.

Klaus Genuit. *Ein Modell zur Beschreibung von Au{ß}enohr{ü}bertragungseigenschaften*. na, 1984.

Google. Daydream View Made by Google, 2016. URL https://vr.google.com/daydream/smartphonevr/https://madeby.google.com/vr/.

Guido van Rossum. Python, 2010. URL https://www.python.org/.

Christian Hoene, Isabel C Patiño Mejía, and Alexandru Cacerovschi. MySofa: Design Your Personal HRTF. In *Audio Eng. Soc. Conv.*, page 6, Berlin, 2017.

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`.

Josef Hölzl. *A Global Model for HRTF Individualization by Adjustment of Principal Component Weights*. PhD thesis, Graz University of Technology, 2014. URL `https://github.com/jhoelzl/HRTF-Individualization`.

S M Hwang and Youngjin Park. HRIR customization in the median plane via principal components analysis. *Proc. AES 31st Int. Conf.*, pages 1–9, 2007. URL `http://www.aes.org/e-lib/browse.cfm?elib=13974`.

Sungmok Hwang and Youngjin Sik Park. HRTF personalization based on artificial neural network in individual virtual auditory space. *J. Acoust. Soc. Am.*, (123), 2008. ISSN 10636676. doi: 10.1109.

Sungmok Hwang, Youngjin Park, and Youn-sik Park. Customization of spatially continuous head-related impulse responses in the median plane. *Acta Acust. united with Acust.*, 96(2):351–363, 2010.

Bob Ippolito. {simplejson}: Simple, fast, extensible JSON encoder/decoder for {Python}, 2017. URL `http://github.com/simplejson/simplejson`.

Eric Jones, Travis Oliphant, Pearu Peterson, and Others. SciPy: Open source scientific tools for Python, 2001. URL `http://www.scipy.org/`.

Gavin Kearney. SADIE — Spatial Audio For Domestic Interactive Entertainment, 2017. URL `https://www.york.ac.uk/sadie-project/binaural.html`.

Frederic L Kistler, Doris J; Wightman. A model of head-related transfer functions based on principal components analysis and minimum-phase reconstruction. *J. Acoust. Soc. Am.*, Vol 91(3):1637–1647, 1992.

Antoni Ligeza. *Artificial Intelligence: A Modern Approach*, volume 9. Prentice Hall, 1995. ISBN 9780131038059. doi: 10.1016/0925-2312(95)90020-9. URL `http://linkinghub.elsevier.com/retrieve/pii/0925231295900209`.

E.˜A. Lopez-Poveda and R Meddis. A physical model of sound diffraction and reflections in the human concha. *Acoust. Soc. Am. J.*, 100:3248–3259, nov 1996. doi: 10.1121/1.417208.

William L Martens. *Principal components analysis and resynthesis of spectral* *cues to perceived direction*. Ann Arbor, MI: MPublishing, University of Michigan Library, 1987.

J.˜C. Middlebrooks and D.˜M. Green. Directional dependence of interaural envelope delays. *Acoust. Soc. Am. J.*, 87:2149–2162, may 1990. doi: 10.1121/1.399183.

John C. Middlebrooks. Individual differences in external-ear transfer functions reduced by scaling in frequency. *J. Acoust. Soc. Am.*, 106(3):1480–1492, 1999. ISSN 0001-4966. doi: 10.1121/1.427176. URL http://asa.scitation.org/doi/10.1121/1.427176.

John C Middlebrooks and David M Green. Observations on a principal components analysis of head-related transfer functions. *J. Acoust. Soc. Am.*, 92(1): 597–599, 1992.

A. Mohan, R. Duraiswami, D.N. Zotkin, D. DeMenthon, and L.S. Davis. Using computer vision to generate customized spatial audio. In *2003 Int. Conf. Multimed. Expo. ICME '03. Proc. (Cat. No.03TH8698)*, pages III–57. IEEE, 2003. ISBN 0-7803-7965-9. doi: 10.1109/ICME.2003.1221247. URL http://ieeexplore.ieee.org/document/1221247/.

Karl Pearson. LIII. On lines and planes of closest fit to systems of points in space. *Philos. Mag. Ser. 6*, 2(11):559–572, 1901. ISSN 1941-5982. doi: 10.1080/14786440109462720. URL `http://www.tandfonline.com/doi/abs/10.1080/14786440109462720`.

F Pedregosa, G Varoquaux, A Gramfort, V Michel, B Thirion, O Grisel, M Blondel, P Prettenhofer, R Weiss, V Dubourg, J Vanderplas, A Passos, D Cournapeau, M Brucher, M Perrot, and E Duchesnay. Scikit-learn: Machine Learning in {P}ython. *J. Mach. Learn. Res.*, 12:2825–2830, 2011.

Project Wilberforce. Final Wireframe, 2017. URL `https://www.assetstore.unity3d.com/en/{#}!/content/81663https://projectwilberforce.github.io/`.

Vikas C. Raykar, Ramani Duraiswami, and B. Yegnanarayana. Extracting the frequencies of the pinna spectral notches from measured headrelated impulse responses. *J. Acoust. Soc. Am.*, 116(4):2625–2625, oct 2004. ISSN 0001-4966. doi: 10.1121/1.4785467. URL `http://asa.scitation.org/doi/10.1121/1.4785467`.

Shoji Shimada, Nobuo Hayashi, and Shinji Hayashi. A Clustering Method for Sound Localization Transfer Functions. *J. Audio Eng. Soc*, 42(7/8):577–584, 1994. URL `http://www.aes.org/e-lib/browse.cfm?elib=6935`.

Chong-Jin Tan and Woon-Seng Gan. User-defined spectral manipulation of HRTF for improved localisation in 3D sound systems. *Electron. Lett.*, 34(25):2387–2389, dec 1998. ISSN 0013-5194. doi: 10.1049/el:19981629.

Unity Technologies. Unity Game Engine. 2017. URL `https://unity3d.com/`.

Peter J M van Laarhoven and Emile H L Aarts. *Simulated annealing*, pages 7–15. Springer Netherlands, Dordrecht, 1987. ISBN 978-94-015-7744-1. doi: 10.1007/978-94-015-7744-1_2. URL `http://dx.doi.org/10.1007/978-94-015-7744-1{_}2`.

Elizabeth M. Wenzel, Marianne Arruda, Doris J. Kistler, and Frederic L. Wightman. Localization using nonindividualized headrelated transfer functions. *J. Acoust. Soc. Am.*, 94(1):111–123, jul 1993. ISSN 0001-4966. doi: 10.1121/1.407089. URL `http://asa.scitation.org/doi/10.1121/1.407089`.

Bosun Xie, Chengyun Zhang, and Xiaoli Zhong. A Cluster and Subjective Selection-Based HRTF Customization Scheme for Improving Binaural Reproduction of 5.1 Channel Surround Sound. In *Audio Eng. Soc. Conv. 134*, may 2013. URL `http://www.aes.org/e-lib/browse.cfm?elib=16780`.

D N Zotkin, R. Duraiswami, L S Davis, A. Mohan, and V C Raykar. Virtual audio system customization using visual matching of ear parameters. *Proc. 16th Int. Conf. Pattern Recognit.*, 3(c):1003 – 1006, 2002a. ISSN 10514651. doi: 10.1109/ICPR.2002.1048207. URL `http://ieeexplore.ieee.org/document/1048207/http://ieeexplore.ieee.org/xpls/abs{_}all.jsp?arnumber=1048207`.

Dmitry N. Zotkin, Ramani Duraiswami, and Larry S. Davis. Customizable auditory displays. *Proc. 2002 Int. Conf. Audit. Disp.*, (July 2-5):1–10, 2002b.