Specification

Automated HRTF Individualisation Based on Localization Errors in 3D Space

Robin Yonge

Febuary 2017

Overview

Virtual reality has come to represent the dominant prediction for the near future of human-computer interaction. If this future is to come to pass, then it is necessary that sufficient consideration is given toward the fidelity of both the visual and the auditory aspects of VR and AR technology. Currently, most implementations of spatial audio in VR and AR applications are based around the use of HRTFs. Most of these implementations use one of a few publicly available datasets, derived from measurements performed on either human subjects or a model such a KEMAR(Algazi et al., 2001)(Gardner and Martin, 1994). For the use of VR/AR to become commonplace the experience must be universal. Because spatial audio relies on complex measurements of real-world subjects (whether or not they are human) applications tend to use average or neutral HRTFs, ones that should work for the largest number of people. Of course, by dint of being average there are many people for whom these do not work. In order to achieve a universal experience, then, it is necessary to find a process for simple individualization of spatial audio that, can be performed by the end user as a part of the standard set-up/calibration process of any VR/AR device or application.

Goals and Research

The goal of this project is to produce a working prototype of an application that takes as input a set of HRTF measurements recorded with a mannequin, and returns an individualised set. For a process like this to achieve widespread implementation it must be simple enough that it can be performed by any given user with nothing more than a head-mounted display, and intuitive enough that it requires of the user no in-depth knowledge of spatial audio. The output HRTF set should also provide a noticeable improvement in the ability of the user to locate sound sources in a virtual 3D environment.

Previous efforts to produce individualized HRTFs through either modification of an existing neutral set or total synthesis have focused primarily on a few different techniques. The most common method is based on a structural HRTF model (Phillip Brown and Duda, 1998; Raykar and Duraiswami, 2003), in which different characteristics of the HRTF are linked to physical characteristics of the pinna, torso, and head. Implementations based on this model have been varied (Tashev, 2014; Xu et al., 2007; Dmitry N. Zotkin, Jane Hwang, Ramani Duraiswami, 2003), but at their core most involve taking measurements of the subjects and attempts to optimise that process. However, the model's reliance on even limited sets of precise anthropometric measurements makes it impractical for my purposes - despite having the potential to be much more efficient than traditional HRIR measurement.

More recent studies have involved applying Principal Component Analysis (PCA) to HRTF data (Hölzl, 2012), in order to identify the spectral features that have the greatest effect upon the incoming sound signal. Once identified, these Principal Components (PCs) can potentially be adjusted in order to produce a more individualised HRTF (Fink and Ray, 2012; Hölzl, 2014). Typically upwards of ninety percent of the information in a given HRTF is based on between 5 and 10 PCs, so through the adjustment of Principal Component Weights (PCWs) it should be possible to individualise HRTFs with a reasonably high degree of accuracy. The combined efficacy and efficiency of this method makes it well-suited to my proposed implementation, and as such this is the method that I will be basing my work upon.

Once they have been identified, I will need data from the user on which to base any adjustment of PCWs. Because of my stated goal of producing an intuitive simple method, I will be performing the individualisation based on a user's attempts to localise sound sources. This method will require me to identify, where possible, points of correlation between PCs and localisation, and as such this will be one of my primary research foci going forward. In order to gather this data from the user it is most practical to derive it from their own localisation errors in a virtual environment, placing the user in virtual space, and asking them to identify the source of the audio cues that will be played to them. This helps to create an intuitive user experience, and giving the user a reticle to point at the perceived source will give precise data upon which to make adjustments.

Based on these decisions, the finished application will be comprised of three parts: The database/files containing the source HRTF data and individualised copies, a front-end interface to be viewed through a head-mounted display, and a module that takes localisation data from the front-end and makes adjustments based on it to the HRTF data.

Technologies

The bulk of the implementation will be written in Python, particularly the core HRTF-adjusting module. Speed isn't too much of an issue as the amount of processing to be done at one time will likely not exceed the capabilities of any device sufficiently powerful to run an HMD, and the access to the SciPy libraries(Jones et al.) will make

reading, visualising, and editing the files containing the HRTF data much simpler. The only minor downside to this approach is that it adds an extra step, processing the initial data, as most of the publicly-available HRTF datasets are stored in .mat files.

For the front-end UI component it is most likely that I will be building it using Unity to allow for quick prototyping and iteration. Other options, be they game engines or frameworks like OpenGL, would only make the process slower and offer no additional benefits - aside from potential efficiency in the case of OpenGL. It's very simple to build a small virtual space around a binocular camera object, adding a component to pass the required data back to the processing module as required. Using a game engine like Unity gives me more options, too, when it comes to processing whatever samples I choose to play to the user. Depending on my needs, I can either rely just on a plugin for Unity, or opt for a dedicated piece of middleware such as FMod or Wwise. Being able to avoid processing the audio myself cuts down significantly on the workload, and potentially means that the final product, or one like it, could very easily be applied to any VR/AR application, provided a comprehensive enough API.

When it comes to testing, the best metric for measuring success is built into the program. The error rates of the user will be recorded in order to make modifications to the PCWs. If stored, this data should be all that is needed to ascertain the overall success or failure of the project. The exact evaluation process has yet to be designed, but a decrease in error rates over time could provide a strong indication toward the efficacy of the algorithm. Through the manipulation of other factors it may be possible rely on this metric with more confidence - for example through the introduction of some degree of chance into the source location or changing room size. If there is little change - or worse, an increase in localisation errors - then the algorithm is flawed. As part of the early investigation into PCA I may also generate a small number of HRIR-derived HRTFs covering a limited number of angles based on measurements of my own head. Though they would not be produced under as rigorous conditions as the neutral set, and would likely not be precise enough to use as proof of success or failure, they may be sufficient enough to serve as a comparison during earlier stages of development.

Timeline

Development Milestones

For the purpose of tracking progress on the implementation side of the project, I think it's helpful to define a set of development milestones.

Basic Functionality:

A set of classes and functions to parse the .mat files that contain the HRTF data and edit raw values as needed, storing the edited version in a copy.

Manual PCW-Adjustment:

Should produce a tool that, based on the functionality implemented already, allows me to manually adjust the weights of specific PCWs. This is essentially what was produced alongside the papers by Josef Holzl(Hölzl, 2014), and should not be too difficult to develop based upon research by him and others.

Algorithm-Complete:

This stage requires me to have at least a preliminary version of the algorithm that I will be trying to use to automate this processing theoretically complete. At this stage I should be able to pass as input an HRTF from a particular angle, along with a vector describing the difference in position between the actual sound source and its perceived location, and receive as output an HRTF that has been modified to take that vector difference into account.

Minimum Viable Product:

The minimum that I could consider as fulfilment of my primary goals; the only difference between this and the complete version of the project is the UI. Requires all of the above features, along with the most basic possible front-end, likely a camera viewer floating in a void with a wire-frame sphere around it to give some visual context.

Finished Product:

The fulfilment of all my stated goals to the best of my ability. As mentioned, the UI difference between this and the above version would purely be the addition of an actual environment, turning the calibration process into almost a minigame. While this isn't strictly necessary to prove that the process is sound, it's important in the context of the project's role as a proof-of-concept, and insofar as I have given placed a degree of importance on the user experience aspects of the project.

Evaluation:

Precise details of the evaluation process have yet to be finalised, but as it will likely involve multiple participants, the use of a controlled environment, and a reasonable amount of analysis after the fact it would be wise to pad the time required somewhat. I should also aim to apply for ethical approval sooner than necessary, so as to avoid any problems on that front.

Submission:

A report should be assembled, covering the project aims, implementation and evaluation process. This report should also detail whether or not the project was a success, based no the data from the evaluation stage. This should be submitted along with supporting materials in the form of the finished product, and the presentation slides that I will be using for the viva.

Research Questions

Research requirements that should be completed prior to beginning implementation work.

Principal Component Analysis:

Review research on Principal Component Analysis, to equip myself with the knowledge required to fully understand how this technique can be applied to my project. As part of this, I should identify the number of PCs that will need to be involved in order to get the best results, and identify correlation between PCWs and localisation errors.

Search Algorithms:

There is a chance existing search algorithms may be helpful in deciding which PCWs to adjust, especially given that in the proposed implementation the individualisation process iterates over multiple steps. Whether or not this is a valuable avenue of research needs to be decided relatively quickly, so as to avoid wasting time.

HRTFs:

More research is needed into the structure of the HRTF data, ways in which the chosen dataset can be modified, and the best ways to visualise it so as to track changes during the development process.

References

V Ralph Algazi, Richard O Duda, Reed P Morrison, and Dennis M Thompson. Structural composition and decomposition of HRTFs. In *IEEE Workshop on the Applications of Signal Processing to Audio and Acoustics*, pages 103–106, 2001. ISBN 0-7803-7126-7. doi: 10.1109/ASPAA.2001.969553.

Larry S. Davis Dmitry N. Zotkin, Jane Hwang, Ramani Duraiswami. HRTF PERSONALIZATION USING ANTHROPOMETRIC MEASUREMENTS.pdf. In *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, 2003.

Kimberly J. Fink and Laura Ray. Tuning principal component weights to individualize HRTFs. In *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 389–392, Kyoto, 2012. ISBN 9781467300469. doi: 10.1109/ICASSP.2012.6287898. URL `http://ieeexplore.ieee.org/stamp/stamp.jsp?tp={&}arnumber=6287898{&}isnumber=6287775`.

B Gardner and K Martin. HRFT Measurements of a KEMAR Dummy-head Microphone. 1994. URL `http://alumni.media.mit.edu/{~}kdm/hrtfdoc/hrtfdoc.html`.

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`.

Eric Jones, Travis Oliphant, Pearu Peterson, et al. SciPy: Open source scientific tools for Python. URL `http://www.scipy.org/`. [Online; accessed 2017-02-22].

C. Phillip Brown and Richard O. Duda. A structural model for binaural sound synthesis. *IEEE Transactions on Speech and Audio Processing*, 6(5):476–488, 1998. ISSN 10636676. doi: 10.1109/89.709673.

V C Raykar and R Duraiswami. Extracting significant features from the HRTF. In *Proceedings of The International Conference on Auditory Display*, pages 115–118, 2003. URL `http://icad.org/Proceedings/2003/RaykarDuraiswami2003.pdf`.

Ivan Tashev. HRTF PHASE SYNTHESIS VIA SPARSE REPRESENTATION OF ANTHROPOMETRIC FEATURES. *2014 Information Theory and Applications Workshop (ITA)*, pages 1–5, 2014. doi: 10.1109. URL `http://ieeexplore.ieee.org/stamp/stamp.jsp?tp={&}arnumber=6804239{&}isnumber=6804199`.

Song Xu, Zhizhong Li, and L Zeng. A study of morphological influence on head-related transfer functions. In *IEEE International Conference on Industrial Engineering and Engineering Management*, pages 472–476, Singapore, 2007. ISBN 1424415292. doi: 10.1109/IEEM.2007.4419234. URL `http://ieeexplore.ieee.org/xpls/abs{_}all.jsp?arnumber=4419234{%}5Cnpapers3://publication/uuid/8C1CB9AC-9961-48AE-A7D1-51270E5E753D`.