Explorations in Spatial Audio and Perception for Virtual Reality

Stanford EE 267, Virtual Reality, Course Report
Instructors: Gordon Wetzstein and Robert Konrad

Nitish Padmanaban
Stanford University
Department of Electrical Engineering
nit@stanford.edu

Keenan Molner
Stanford University
Department of Electrical Engineering
kmolner@stanford.edu

Abstract

A great deal of the Virtual Reality research and implementation is focused on the graphics and hardware associated with immersive virtual reality experiences. To further enhance the VR experience, we wanted to explore the ways in which audio contributes to the immersive nature of the medium. Specifically we, investigated how the perceived audio in a virtual environment changes with knowledge of the position of the listener across a wide range of frequencies, implemented with a Head Related Transfer Function (HRTF). Furthermore, we built a virtual experience to learn about audio perception, like windowing, the Fourier domain, and audio processing. We use the IMU sensor data to enable interaction with the audio processing in our scene, like a virtual reality version of the Chrome Music Labs.

1. Introduction

Our main focus for this project was to build an interesting, interactive environment for Virtual Reality with a main focus on audio perception and signal processing. Most audio experienced by a consumer through a headset is divided into Left and Right audio tracks. When a sound is intended to be coming from the left side of the user, the audio is mainly played from the left headphone, with a small amount of the signal played through the right headphone. This makes the source sound closer to the left side of the user, but still contained along the line between the two ears in the user's head. This doesn't feel nearly as immersive as the real world it aims to replicate.

1.1. The KEMAR Dataset

For Virtual Reality, this left/right audio panning doesn't make the user feel like they're immersed in the world visualized on screen. In order to the visuals of the world to match the soundscape, designers must take the geometry and position of the user's head into account. In 1994, Gardner and Martin constructed the KEMAR dataset which records the transfer functions in response to broadband impulses spaced at 710 different directions around the head, for both the left and right ears [1]. Using this set of impulse responses, we can construct audio filters for both the left and right ear to sound perceptually similar to spatially localized audio in the real world. A single sound source $x(t)$ at a point in 3D space can produce two seemingly spatial audio signals for the left and right ears, $x_L(t)$ and $x_R(t)$ by multiplying the incoming signal by the HRTF transfer functions, $H_L(\omega)$ and $H_R(\omega)$, respectively, in the frequency domain. The resulting two audio signals now have the same delay and frequency response appropriate for each ear from the recorded data on the KEMAR dummy.

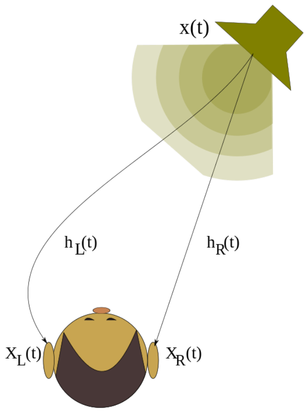

Figure 1. HRTF Signal Synthesis [2]

2. Related Work

In 2003, Harma et. al. presented a framework for real-time, wearable, augmented reality audio processing [3].

They attached two small microphones to the ear to collect and process the sounds of the natural environment. On top of these natural sounds, the researchers add artificial sounds to create the augmented reality environment, processed through a position dependent HRTF. The combination of the natural sounds collected from two different sources overlaid on top of the augmented environment makes the augmented sound effects sound more intertwined and embedded in the natural world. They confirm this assertion with user studies, in which test subjects were unable to distinguish sound effects played through the headphones from sounds coming from the natural world around them. These tests assert the importance of a proper HRTF when building a virtual or augment environment.

Sundareswaran et. al. constructed a user-guidance system using 3D audio cues and ran user studies to determine the effectiveness of 3D audio to locate the position of audio sources in the virtual world. Localizing a single point audio source was most difficult behind the head of the user. Before using 3D audio, test subjects were able to characterize the location of an audio source within only 40 degrees. With the incorporation of 3D audio, the accuracy of localization was reduced to within less than 25 degrees, suggesting an almost twofold increase in localization ability [4].

3. Project Overview

Our project underscores the importance of proper spatial audio implementation and provides a learning environment for users to experience audio filtering, pitch shifting, windowing, and real-time audio processing.

3.1. Real Time Audio Processing

To accomplish real time audio processing, our program reads mono-channel audio data into the processing pipeline every 4096 samples, or about every 10ms. Each buffer of 4096 samples is then multiplied by an MLT Sine Window

$$x_{\text{analysis}}[n] = x_{\text{mono}}[n] \times w_{\text{rect}}\left[\frac{n}{M}\right] \times \sin\left[\left(n+\frac{1}{2}\right)\frac{\pi}{2M}\right]$$

where M is the length of the sample buffer – 4096 samples in our case [5].

Our windowed sample buffer then undergoes a Discrete Cosine Transform, which we've implemented with the Exocortex DSP library's real-valued Fast Fourier Transform [6]. Our windowed samples now reside in the form

$$X[\omega_k] = DCT\{x_{\text{analysis}}[n]\}$$

which give us a symmetric signal about the $\omega_k = 0$ axis and extends for M samples in either direction. We only care about the positive frequencies, so we reduce our data to only contain these samples. With the frequency domain representation of these samples, we're able to apply our signal processing techniques easily.

3.2. Pitch Shifting

For one of the effects position dependent that we applied to the audio, we chose to do a pitch shift when the head was rotated about the z axis. However, unlike normal pitch shifting, we shift circularly: the highest frequencies – when shifted past the range of hearing – wrap back into the lowest frequencies, and vice versa.

There are two ways to approach shifting of frequencies. One is to simply shift frequencies linearly. While this is easy, human auditory perception, much like other aspects of perception such as vision, is logarithmic in nature. There are ways to adapt the Fourier Transform to this, such as implementing a constant Q transform using the FFT [7], using a logarithmic Fourier Transform, or binning the FFT with logarithmic spacing.

The constant Q transform's treatment of lower frequencies is such that they have less temporal resolution than higher frequencies. Since our intent was to pitch shift using frequency domain methods, there was some concern that by shifting frequencies, we would effectively be reducing the time resolution of the input, especially when shifting the highest input frequencies to the lowest output. This would effectively render the extra time resolution of the higher frequencies useless, since they would be limited by the lowest ones. The logarithmic Fourier Transform on the other hand required logarithmically spaced time samples, which did not seem ideal given our desire to invert the transform and change output in realtime.

In using the third approach of binning the FFT coefficients, we grouped them into the same 10 octaves as our output sources. When shifting, we needed to maintain the same amount of energy in each octave. When shifting to higher frequencies, we approximated this by duplicating frequency components the necessary amount of times to fill the linearly wider spectrum, and at lower frequencies, we added adjacent indices and applied a scaling factor. This was tested empirically using a pure tone at 440Hz and comparing the output waveform's relative amplitude at different pitch shifts.

3.3. Spatialized Audio Sources

For this project, we needed a 3D audio source. In order to use the rich library of audio that already exists, we chose to create such sources by distributing the frequency components of the audio around the user's head, with low frequencies starting behind the user and wrapping around the head, clockwise, as the frequency increases. To accomplish this, we divided our transformed samples into 10 different bins – again, logarithmically, to perceptually contain the same shift in frequency between each audio source. The normalized discrete frequencies, $k\frac{\pi}{M}$, are therefore binned as k = 0–7, 8–15, 16–31, 32–63, 64–127, 128–255, 256–

511, 512–1023, 1024–2047, and 2048–4095.

To accomplish this binning, we copy the sample points of the full spectrum across the indices of interest into a sample buffer of zeros, essentially nulling out all other frequencies we aren't interested in. The result of this step is 10 discrete bins of frequency data, each logarithmically spaced in length.

3.4. Frequency cutoff

As a way to allow the users to explore different frequencies of a complex sound such as a song, we implemented a method of reducing the number of frequencies played based on head rotation about the x axis. As the user tilts their head upward, a wider spectrum of the song can be heard, and down to only a single audio source as they tilt downwards. This single central audio source is chosen based on the source the user is looking at (i.e. y axis rotation).

In this way, the degree of forward tilt changes the degree to which the user wants to explore the spatial aspect or the frequency aspect of their sound. In another context, with a richer 3D audio environment, this is could be thought of as a way to "focus" on a single direction of interest after identifying the direction in a full 3D spatial mode.

3.5. Synthesis Windowing and COLA

Taking the Inverse Fourier Transform of the 10 discrete frequency buffers results in 10 separate audio tracks, each containing data within a specific frequency band.

Once again, we multiply our resulting audio samples by the MLT Sine Window. By using an MLT Sine Window for both analysis and synthesis, we effectively implement a single pass Hann window, given by the equation

$$w_{\text{Hann}}[n] = w_{\text{rect}}\left[\frac{n}{M}\right] \times \cos^2\left(\frac{\Omega_M}{2}\pi\right)$$

where M is the window length and Ω_M is the main lobe bandwidth $\frac{2\pi}{M}$.

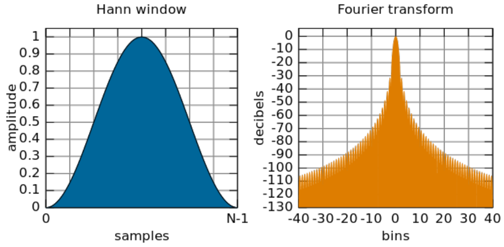

Figure 2. Hann Window

The Hann window offers a -13dB sidelobe rejection and is easy to implement in software. Additionally the Hann window has a minimum Constant Overlap and Add property (COLA) of $R = \frac{M}{2}$, which makes real-time output streaming easy, since the samples must overlap by a minimum of R to achieve a constant output signal amplitude after processing.

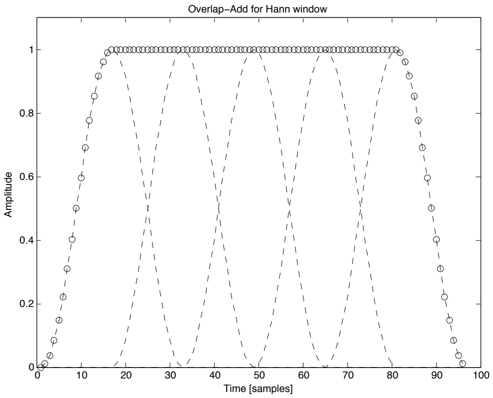

Figure 3. COLA overlap

After filtering the output sample, we place the sample into its respective output buffer with a COLA step of $R = \frac{M}{4}$. We attach each audio buffer to a separate sound player object in Unity, distributed uniformly on a circle about the user.

3.6. HRTF

The HRTF we chose to implement for our demo comes from the Unity Audio Spatializer SDK from the Native Audio Plugins package[8]. This spatializer implements the HRTF based on the KEMAR dataset. When we initialize all audio sources in the world, we enable spatialization of the source, allowing it to be fed into the HRTF.

To underscore the importance of the HRTF and its perceptual differences between simple Left/Right panning, we enabled the user to switch the audio processing mode. By holding down the "F" key, all spatialization is eliminated and the user hears the normal Left/Right mix. It becomes much more difficult to locate the source of the audio without the HRTF.

3.7. Visual World

Our graphics in the 3D world aim to visualize the audio heard in the world. The sky sphere encompassing the entire world is a visual spectrum, with the low frequency color, red, placed next to the speaker object playing the lowest frequency audio. The spectrum then wraps around the world with increasing light frequencies mapping to the increase in audio frequencies played by each speaker object. The spectrum wraps back into itself between the highest and lowest audio bins in the back of the head. A side to side rotation of the head allows the user to see the different colors of the spectrum. An up/down tilt changes the brightness of the world, from black at the bottom to white at the top.

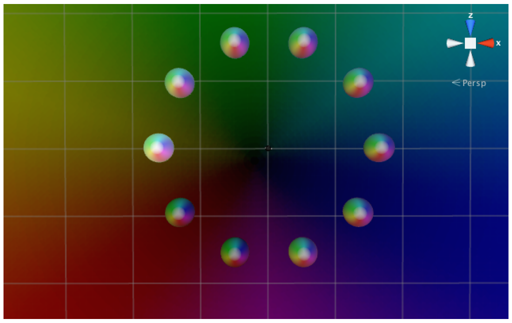

Figure 4. Bird's eye view of virtual world

Each sound object in our virtual world is contained in a metallic sphere, reflecting the rainbow world around it. The brightness parameter of the metallic material is controlled by the amplitude of the audio playing out of the speaker object. The louder the amplitude, the brighter the ball, creating a visual analog to the multi-band audio the user hears and reinforcing the spatial location of the audio source.

4. Results

When compared with stereo audio, the spatial localization afforded by use of the HRTF was immediately apparent to users. In particular, use of stereo audio causes the perception that the audio is coming from a line connecting the ears inside one's head. The HRTF allowed us to localize the sounds to a ring around the user, but in theory, to anywhere.

However, there were some limitations where it becomes harder to localize, especially with sound behind the user. This may be due to the fact that the KEMAR data is a general dataset collected using a model head, whereas the true HRTF varies slightly for every person.

5. Future Work

This project demonstration was primarily built for entertainment purposes and to serve as a virtual reality version of the Chrome Music Lab. We believe the audio and visuals provide a good example of Fourier analysis, windowing, pitch shifting, transfer functions, and real-time audio processing. This demonstration is a good educational tool for first introductions to these concepts and showcases the effect and importance of spatial audio in virtual reality. In addition, we also believe that our demonstration can be used as a form of user gaze guidance in a virtual world to effectively tell a story. Work to better understand and characterize possible positions for reconstructive audio interference and nulling when using the HRTF also proves to be an area of further research and development.

5.1. Gaze Guidance

There are many methods of gaze guidance. One way is to simply have an arrow near the edge of a screen pointing in the direction to turn. People also tend to look at things that move or make sound. The advantage of using sound here is twofold. First, unlike arrows, using well thought out sound doesn't obscure the field of view. Second, though one could guide the user with subtler visual cues than arrows by weaving it into the visual story, it's harder to guarantee that it actually is seen by the user; we can't see all around us, but we can hear from any direction.

This means that an effective use of audio as gaze guidance, either instead of or in addition to visual guidance, could be a lot more powerful than visuals alone. This is especially true given the ability of people to pinpoint the exact direction of sound when using an HRTF and 3D audio. However, this would still be valid, if somewhat more limited, with stereo audio: one would be able to guide the user left and right, but never up and down. Unlike both visual guidance and stereo, 3D audio presents the possibility of simply identifying the correct direction and looking straight there, instead of slowly being guided towards it.

5.2. Audio Nulling of HRTF

The HRTF in the KEMAR dataset is a collection of impulse responses for each ear from 710 discrete locations. When people move their head through a continuous space, a choice must be made on how to localize the audio. One choice is to simply choose the impulse response recorded from the nearest location to that of the source and play it from there. The particular dataset used would ideally optimize the locations used for the impulse responses such that there is a higher resolution in directions that are also perceptually easier to differentiate.

The other approach is to interpolate intelligently between two impulse responses. This can sometimes lead to undesirable effects wherein the impulses combine in such a way as to completely null out certain frequencies or ranges in some interpolated regions, even though a real sound would not share the same dropped frequencies. There are ways to potentially mitigate this effect, such by using minimum or linear phase HRTF impulses; but whichever of these achieves the most realistic experience while trading off against latency will inform decisions on how to approach expanding HRTFs to continuous space.

6. Acknowledgments

We thank Prof. Gordon Wetzstein and Robert Konrad for their support and teaching.

References

[1] Bill Gardner, Keith Martin, et al. Hrft measurements of a kemar dummy-head microphone. 1994.

[2] Head-related transfer function.

[3] Aki Harma, Julia Jakka, Miikka Tikander, Matti Karjalainen, Tapio Lokki, and Heli Nironen. Techniques and applications of wearable augmented reality audio. In *Audio Engineering Society Convention 114*. Audio Engineering Society, 2003.

[4] Venkataraman Sundareswaran, Kenneth Wang, Steven Chen, Reinhold Behringer, Joshua McGee, Clement Tam, and Pavel Zahorik. 3d audio augmented reality: implementation and experiments. In *Proceedings of the 2nd IEEE/ACM international symposium on mixed and augmented reality*, page 296. IEEE Computer Society, 2003.

[5] Julius O. Smith. *Spectral audio signal processing*. W3K, 2011.

[6] Ben Houston. Exocortex.dsp. *Exocortex.DSP*, Oct 2003.

[7] Judith C Brown and Miller S Puckette. An efficient algorithm for the calculation of a constant q transform. *The Journal of the Acoustical Society of America*, 92(5):2698–2701, 1992.

[8] Audio spatializer sdk, 2016.