Explorations in Spatial Audio and Perception for Virtual Reality

Nitish Padmanaban, Keenan Molner
Department of Electrical Engineering, Stanford University

Motivation

While the majority of the EE267: Virtual Reality course focused on the graphics and hardware associated with immersive virtual reality experiences, we wanted to explore the ways in which sound contributes to the immersive nature of the medium. Specifically, we aim to investigate how the perceived audio in a virtual environment change with knowledge of the position of the listener across a wide range of frequencies. Furthermore, we will build a virtual world that allows the user to discover the tradeoffs between using different audio models and an experience to learn about audio perception, like windowing, the Fourier domain, and audio processing. We will use the IMU sensor data enable interaction with the audio processing in our scene, as well.

Technical Implementation

In order to create the feeling of real-time audio processing, we read the data from the file, process it, and store it into a buffer only a few frames before the playback reads from the buffer. Each frame of 4096 samples is windowed by an MLT Sine Window in the time domain. After windowing, we compute the Fourier transform of the signal.

$$X[\omega] = FFT(x[n] * w_{rect}[n] * cos(\frac{\pi n}{4096}))$$

Frequencies in X[ω] are then binned logarithmically into 10 separate frequency bands. The pitch angle of the user's head changes the cutoff frequency of the samples in the buffer, resulting in a downward tilt limiting the audio to a single band and an upward tilt playing all bands. The cutoff filtering is equivalent to changing the length of the analysis window:

$$win[n] = w_{rect}(\frac{n}{head\ tilt}) \times cos(\frac{\pi n}{head\ tilt})$$

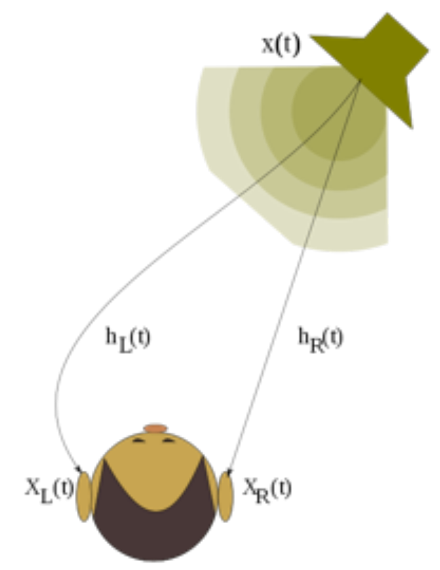
A side to side rotation of the head pushes the Fourier samples in one bin into the adjacent bin before taking the IFFT, resulting in a perceptual pitch shift. Finally, each frequency bin was converted back into a time-domain sample bin and windowed by the same MLT Sine Window as before, achieving a total analysis and synthesis window equivalent of the Hann window.

$$for\ j = (1 : num\ bins)$$
$$x_j[n] = IFFT(X_j[\omega] * w_{rect}[n] * cos(\frac{\pi n}{4096})))$$

Our Virtual Experiment

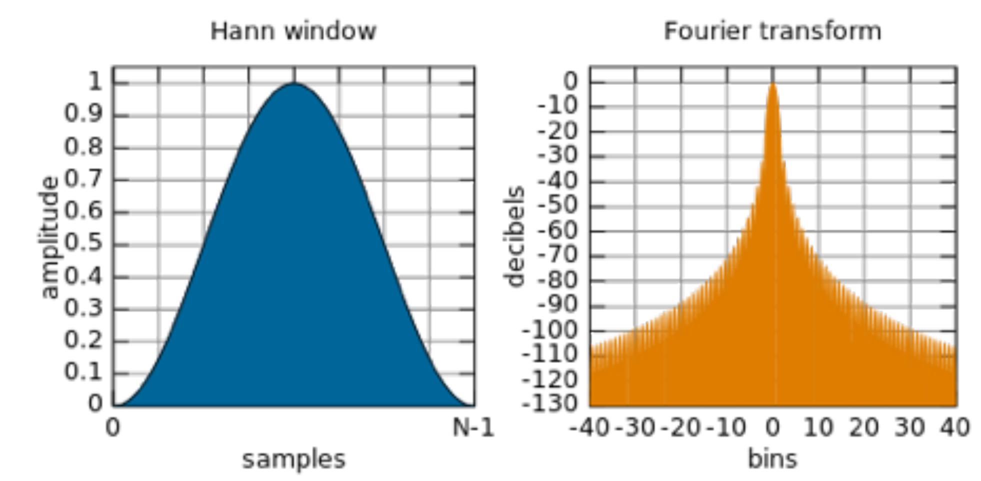
HRTF

- Implemented the HRTF from the KEMAR Dataset.
  - Describes impulse response for all points in space surrounding the head.
  - Makes audio spatial, rather than existing inside the user's head.
- Contrast this with the typical L/R model for audio.



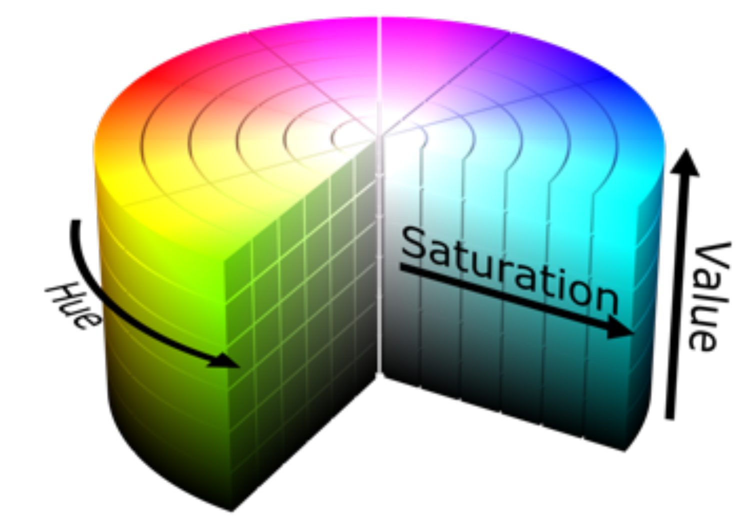
Signal Processing

- Real-time audio stream and processing.
- FFT taken of each 10ms sample.
  - Sound is binned by frequency.
  - Pitch shifted and bin cutoff changed by user head rotation.



Visuals

- Visual spectrum represents the low to high frequencies of sound with different light frequencies.
- Brightness changes with head rotation.
- Balls represent each audio frequency band, glowing in response to audio amplitude at that frequency.



Uses for Our Work

User Gaze Guidance

- With the proper implementation of the HRTF, it becomes very easy to localize the source of audio in space.
- Without any visual cues, it's very easy to point in the direction of the audio source.
  - ex: chirp file
- Localizing the sound direction is much easier by using the HRTF, suggesting that a user's focus and attention in a virtual world could be captured by localizing the loudest sound to a specific point in space with the HRTF.

Virtual Chrome Music Lab

- Google Chrome Music Lab example for virtual reality.
- Allows students to explore audio filtering, synthesis, and analysis in the virtual space.
- As gesture recognition and hand interaction becomes more common in VR, students and artists alike can begin to synthesize music for virtual reality using similar techniques

References

- Gardner, B. and Martin, K. 1994. HRTF Measurements of a KEMAR Dummy-Head Microphone.
- HSV Cylinder. 2010. wikipedia.org.
- Hann Function. 2013. wikipedia.org.
- HRTF. 2007. wikipedia.org.
- ROYALTY FREE MUSIC by BENSOUND. Royalty Free Music by Bensound. http://www.bensound.com/.
- Smith, J.O. 2015. The MLT Sine Window. Spectral Audio Signal Processing. https://ccrma.stanford.edu/~jos/sasp/mlt_sine_window.html.
- Unity Audio Spatializer. 2016. Unity Audio Spatializer SDK. http://docs.unity3d.com/manual/audiospatializersdk.html.
- Unity Technologies. 2016. NativeAudioPlugins. Native Audio Plugins. https://bitbucket.org/unity-technologies/nativeaudioplugins
- Unity. Sound Effects & Scripting. https://unity3d.com/learn/tutorials/topics/audio/sound-effects-scripting.
- Tikander, M., Harma, A., and Karjalainen, M. Binaural positioning system for wearable augmented reality audio. 2003 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (IEEE Cat. No.03TH8684).