Audio Engineering Society

Convention e-Brief 400

Presented at the 143rd Convention
2017 October 18–21, New York, NY, USA

Audio Localization Method for VR Application

Joo Won Park[1]

[1] *Columbia University*

Correspondence should be addressed to Joo Won Park (`jp3378@columbia.edu`)

ABSTRACT

Audio localization is a crucial component in the Virtual Reality (VR) projects as it contributes to a more realistic VR experience to the users. In this paper, a method to implement localized audio that is synced with user's head movement is discussed. The goal is to process an audio signal real-time to represent three-dimensional soundscape. This paper introduces a mathematical concept, acoustic models, and audio processing that can be applied for general VR audio development. It also provides a detailed overview of an Oculus Rift- MAX/MSP demo.

1 Introduction

This paper introduces a method to localize audio in Virtual Reality (VR) application. This paper uses MAX/MSP as the VR platform—as the front-end development that brings processed audio and VR display in Oculus Rift together. A particular audience is people who intend to localize audio in their MAX/MSP VR projects so that the audio environment is synced with the VR user's head movement in an Oculus Rift-MAX/MSP setup. Extended audience is people who seek general method to easily implement user-synced audio in some VR platform. This paper covers an essential mathematical concept, quaternion, as well as mathematical modeling that creates a sense of three-dimensional (3D) auditory space.

There are three parts to this paper: The first part is on acoustic modeling that creates three-dimensional auditory dimension. Methods to model Inter-aural Level Difference (ILD) and Head-Related Impulse Response (HRIR) convolution and interpolation are introduced. Note that the model is simplified by limiting rotation to the yaw axis on the horizontal plane. Such simplification allows easier quaternion algebra and acoustic modeling while it can be extended to rotation with elevation angle. The second part covers actual implementation in coding by using quaternion algebra. The last part presents the Oculus Rift-MAX/MSP VR demo as an example of a user-interactive audio environment in VR that uses the methods introduced in the paper.

2 Acoustic Modeling

Interaural Level Difference (ILD) and Interaural Time Difference (ITD) are the differences between the two ear signals that are most relevant for the localization of sound source on the horizontal plane.[1] HRIR's of the two ears describe this difference, thus serving as the cue for the sound location in terms of the azimuth angle.

The sound in VR should accurately represent the change of the distance between the user's ears and the sound source–the further the user is from the sound source, the softer the sound should be due to the spreading loss.[2] HRIR's are only measured uniformly at 1 meter away from the sound source, so an additional model is needed to reflect the sound level change by the distance beyond 1 meter.

The following mathematical functions are acoustic models of sound level by distance. The sound level decreases non-linearly, as spherical spreading causes the level to decay much rapidly when further.[2] Logarithmic function is applied for level decay in shorter distance (distance ≤ 15) to have a concave down function, and inverse functions is applied for longer distance (distance > 15) to have a concave up function. Note that constants in these models must be adjusted accordingly to the VR development environment, as well as by the adequate judgement of "closeness". In the demo for this paper, 15 is the unit length in the Oculus Rift-MAX/MSP setup, and the constants are determined accordingly. The constants of the logarithmic and inverse functions are adjusted as in equations 1 and 2. Figures 1 and 2 summarize that the sound level drops mildly in closer distance, and in longer distance the level drops more rapidly. $0 < f(x) < 1$ represents the sound level decrease, and x represents the distance between sound source and the listener.

* Short Distance

$$f(x) = .33 * \log(22 - 1.4 * x) \tag{1}$$

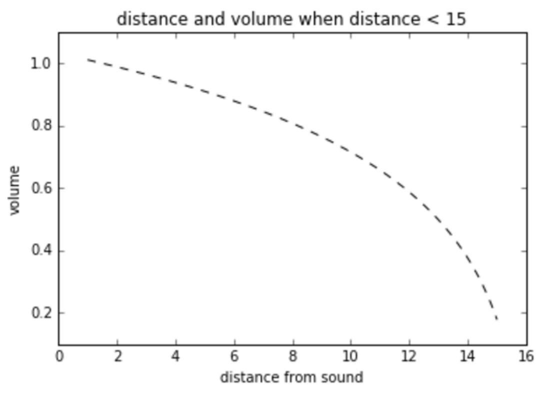

Fig. 1: sound level decay in short distance

* Long Distance

$$f(x) = \frac{1}{x} \tag{2}$$

3 Implementation

A 16 seconds long drum loop (mono) is used for the demo of this paper, but it can be substituted by other audio sample of choice. The audio sample is then convolved with Head-Related Impulse Response (HRIR).

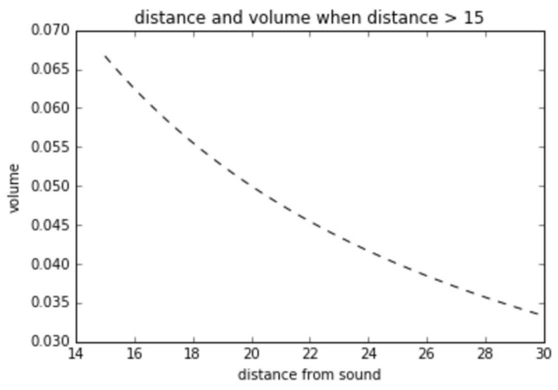

Fig. 2: sound level decay in long distance

Sets of HRIR data were chosen from New York University's Music and Auditory Research Laboratory (MARL) [3]. The python notebook script that covolves an audio file with a selected HRIR data set can be found in github [4].This python script extracts 24 HRIR's on horizontal plane (0 elevation, 15° azimuth increment) and convolves them with the loaded audio file. Then, it creates 24 audio files from the convolution that are of equal loudness. These 24 audio files will be the ingredients for designing 3D auditory scene, and are saved in the local directory.

Javascript is used for quaternion computations to process orientation information fed from the Oculus Rift. The javascript is implemented in MAX/MSP where a set of HRIR-convolved audio files is weighted accordingly to the user's orientation.

3.1 Quaternion Algebra

Quaternion is a mathematical concept similar to imaginary numbers, and it is an integral part of representing user's head orientation and thus each ear's location after user's head movement. This section illustrates the concept of quaternion, its properties, and how it is applied to the tasks of this project.

The idea of quaternion is first described by William Rowan Hamilton [5]. A quaternion is represented four real numbers, say q_1, q_2, q_3, q_4, and imaginary units, $\hat{\imath}, \hat{\jmath}, \hat{k}$. A quaternion $q = q_1\hat{\imath} + q_2\hat{\jmath} + q_3\hat{k} + q_4 = (q_1, q_2, q_3, q_4)$ represents a rotation if a quaternion q can be expressed as follows[6]:

$$q = v_x \sin\frac{\theta}{2}\hat{\imath} + v_x \sin\frac{\theta}{2}\hat{\jmath} + v_x \sin\frac{\theta}{2}\hat{k} + \cos\frac{\theta}{2}$$

$v = (v_x, v_y, v_z)$ represents a unit vector along the axis of rotation, and θ an angle of rotation. q is called "quaternion of rotation".

This concept useful because Oculus Rift produces positional values and quaternion values that correspond to user's head rotation along the yaw axis. This allows real-time computation of ear's location and angle of rotation. The computation of ear's location is used for calculating the distance between each ear and the sound source, and computation of the angle of rotation is used for weighting convolved audio samples from the python script.

Depending on the environment the developer is working on, the definition of angle θ is adjusted. In this paper and the demo, the angle of rotation θ is the clockwise rotation angle from the $-z$ axis. Also, the position of the sound source is fixed on xz plane. The number that represents user's head diameter (2.0) is arbitrary for the ease of computation, and it is assumed that the length from the center of the head to each ear is 1.0.

Limiting to yaw axis rotation simplifies the quaternion values. Axis of rotation is the y axis, so $v = (0,1,0)$. Consequentially, $q = (0, \sin\frac{\theta}{2}, 0, \cos\frac{\theta}{2})$. Oculus Rift's head tracker returns positional values x, y, z and quaternion constituents q_1, q_2, q_3, q_4 through MAX/MSP. In the simplified task limited on the xz plane and the yaw axis rotation, only x, z positional values and $q_2 = \sin\frac{\theta}{2}$, $q_4 = \cos\frac{\theta}{2}$ orientation values are relevant.

3.2 Distance Computation

Given user's positional value (x_0, z_0) and quaternions (q_2, q_4), I calculated each ear's position. As suggested in Figure 5, location of the ears given head's center position (x_0, z_0) is $(x_0 - \cos\theta, z_0 + \sin\theta)$ for the left ear, $(x_0 + \cos\theta, z_0 - \sin\theta)$. Due to the properties of quaternion of rotation, $\cos\theta$ and $\sin\theta$ can be expressed in terms of quaternions' constituent:

$$\sin\theta = 2\sin\left(\frac{\theta}{2}\right)\cdot\cos\left(\frac{\theta}{2}\right) = 2q_2q_4$$

$$\cos\theta = \cos^2\left(\frac{\theta}{2}\right) - \sin^2\left(\frac{\theta}{2}\right) = q_4^2 - q_2^2$$

Thus, each ear's location can be calculated real-time as Oculus Rift's headset gear returns positional values and quaternions. This allows to calculate the distance between the sound source and each ear, which the acoustic model from Section 2 takes as input x.

Location of each ear after θ angle of rotation:

* Left ear: $(x_0 - q_4^2 + q_2^2, z_0 + 2q_2q_4)$

* Right ear:$(x_0 + q_4^2 - q_2^2, z_0 - 2q_2q_4)$

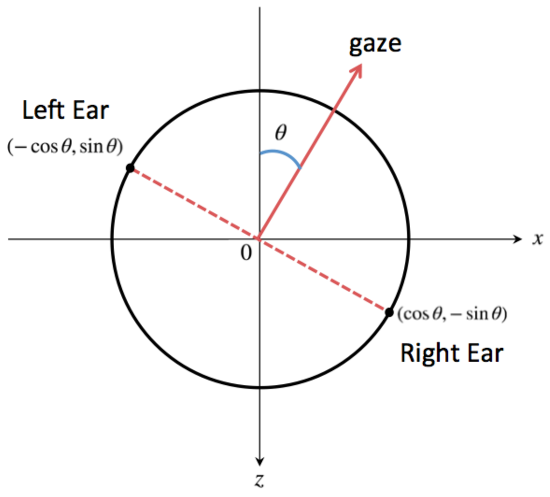

Fig. 3: Location of each ears

3.3 Interpolation

24 audio files are created from the python script [4]. These files are drum sample convolved with HRIR at angles in 15° increment. On the xz plane, when angle of rotation is exactly $15°, 30°, ..., 345°$, simply playing corresponding convolved audio file would be accurate representation of the localized audio in VR. However, for angles that are not exactly in 15° increments, interpolation is necessary. Figure 4. describes the algorithm of weighing two audio files to interpolate localized audio for any angle of rotation.

*Algorithm:

1. Divide parametric space $(0 \leq \theta \leq 360°)$ into 24 bins (each bin is of 15° increment).

2. Compute the angle of rotation θ from the quaternion values returned from Oculus Rift:

$$\theta = 2\cos^{-1} q_4$$

Note that there are two possible values of θ from the inverse cosine function. It is necessary to compare the two possible options' half angle sine values and pick the one that is closer to $q_2 = \sin(\theta/2)$

3. Determine which bin the computed angle of rotation (θ) belongs to.

4. Interpolate audio signal as a mix of two audio signals that bounds the bin. For example, in Figure 6., at angle of rotation between 15° and 30°, the audio signal should be mix of File 2 (HRIR convolved for 15°) and File 3 (HRIR convolved for 30°). If θ is closer to 15°, File 2 should dominate, and vice versa. The following is the exact computation:

Given θ and Bin B_n given, the interpolated audio signal S should be mix of audio signal of the files that bound the bin, S_n and S_{n+1}. Weights for the other audio signals are 0.

$$S = (n+1-\frac{\theta}{15^\circ}) \cdot S_n + (\frac{\theta}{15^\circ} - n) \cdot S_{n+1}$$

The javascript code that is implemented in the demo can be found in github [7].

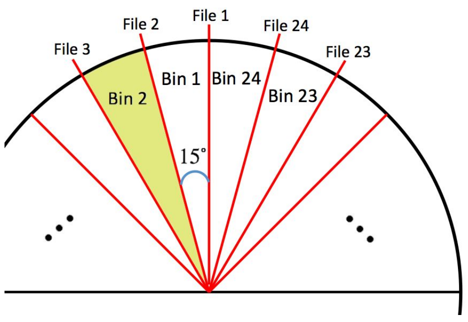

Fig. 4: Algorithm for Weighting Audio Signals

4 Demo

MAX/MSP is used as a bridging front-end platform that receives locational information from the Oculus Rift headset, manipulates the loaded audio signals accordingly to that information, and outputs the interpolated audio signal as well as the visual display to the headset. Professor Bradford Garton's MAX/MSP patch [8] run simple visual display and receives Oculus Rift's locational data. The demo is built over this patch as a basis. Javascript codes are implemented to process the loaded audio signals (HRIR convolved drum samples) to output interpolated audio signals, and MAX/MSP synchronizes the visual display with the audio signal.

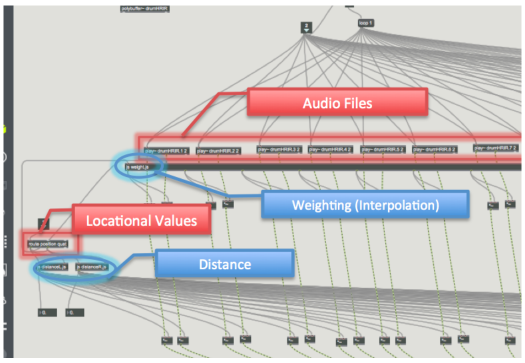

Fig. 5: MAX/MSP Patch

Figure 5. is a part of the MAX/MSP patch that manipulates and processes the audio signals. This patch receives user's locational data from Oculus Rift headset and plays 24 HRIR-convolved audio files simultaneously (both marked in red boxes in Figure 5.). These two are used as inputs that are processed with javascript codes that interpolate the audio signals and calculate the distance between the sound source and the user's ears (both marked in blue boxes in Figure 5.).

4.1 MAX/MSP Demo Directions

This section describes how to use the Demo. The objective of this demo is to play a desired audio sample (mono signal) and manipulate it to create an auditory scene that is synchronized with user's position and orientation in VR. The following is the instruction of the demo:

1. Create 24 HRIR-convolved audio files using the python script[4] for the desired audio sample

2. Load the folder with the audio samples into MAX/MSP's polybuffer object.

3. Wear the Oculus Rift headset and earphones, and start the program. Toggle fullscreen.

4. Navigate in VR using the keypads and by moving the head.

* key commands:

- w/up arrow: move forward
- s/down arrow: move backward
- d: move right
- a: move left
- right arrow: rotate right
- left arrow: rotate left
- delete: reset
- escape: toggle fullscreen

5 Summary

The task of this paper is to interpolate HRIR-convolved audio signals to recreate realistic auditory environment in VR when user's head movement is limited to yaw rotation. A simple mixing by weights method was used to interpolate for angles of rotation that were not strictly at 15° increments.

This project can serve as a basis framework to develop realistic auditory environment in VR. Some adjustments that can be made are the choice of HRIR data set (which HRIR data set optimizes the accuracy?), and re-assessment of acoustic models (how does the distance and direction affect sound perception?). This project can be extended beyond the yaw rotation limitation by employing an appropriate quaternion algebra.

Another important task to be solved is to develop a method to evaluate the accuracy of the auditory scene created in the demo. For this project, I used my own subjective judgment to assess if the recreated auditory scene was "good enough". But for accurate test procedures, an objective metric for assessing the auditory scene created (interpolated audio signal) is necessary.

6 Acknowledgements

I would like to thank Professor Nima Mesgarani and Professor Bradford Garton of Columbia University for their guidance and helpful advice.

References

[1] Raspaud, V. H., M. and Evangelista, G., "Binaural Source Localization by Joint Estimation of ILD and ITD," *IEEE Transactions on Audio, Speech, and Language Processing*, 18, pp. 68–77, 2010.

[2] B, T., *Handbook for Acoustic Ecology*, World Soundscape Project, Simon Fraser University, and ARC Publications, 1978.

[3] Andreopoulou, A. and Roginska, A., "Documentation for the MARL-NYU file format Description of the HRIR repository," 2011, data Retrieved from NYU Music and Audio Research Laboratory.

[4] Park, J. W., "VR Audio," 2017, github.

[5] Hamilton, R., "On Quaternions, or on a New System of Imaginaries in Algebra," *Philosophical Magazine*, 1850.

[6] Trawny, R. S., N., "Indirect Kalman Filter for 3D Attitude Estimation," *Multiple Autonomous Robotic Systems Laboratory*, 2005.

[7] Park, J. W., "weight," 2017, github.

[8] Garton, B., "Oculus Rift," 2016, website.