IMPLEMENTATION OF 3D AUDIO USING INTERPOLATED HEAD-RELATED TRANSFER FUNCTIONS

Michael Heilemann, Kedar Shashidhar, Alexander Venuti

University of Rochester Dept. of Electrical and Computer Engineering

ABSTRACT

Humans are able to interpolate changes in sound pressure in order to infer the spatial location of the sound source. The simulation of location-based sound is known as binaural audio. There are several challenges to accurately representing a 3D sound field in headphones. The first of which is limiting the amount of measured impulse responses needed. Having to store an impulse response for each point in space can take up a great deal of memory. The second challenge is creating a smooth transition between points as location changes. We have investigated the extent to which the number of stored impulse responses can be reproduced. By using a time domain interpolation method, we have reduced the number of necessary points from 1250 to 91, while keeping the transition smooth from point to point. We also investigated how 3D HRTF interpolation can be implemented in real-time.

1. INTRODUCTION

A Head-Related Transfer Function (HRTF) is a location-dependent function that models the spectral changes occur when sound travels from its source to the eardrum. The particular value to an HRTF is dependent upon both the location of the source relative to the listener as well as the physical size and shape of the listener. When a mono sound source is filtered by a specific location's left and right HRTFs, the resulting stereo sound is experienced as if being generated from that location in space. Due to this property of HRTFs, the virtual spatialization effect is best experienced when the filtered audio is directly sent to the listener's ears, or in other words the listener uses headphones.

The HRTF Library used in this project is provided by UC Davis' Center for Image Processing and Integrated Computing (CIPIC) [1]. Their database consists of 1250 different locations using 25 different azimuth indices and 50 different elevation indices for 45 different subjects.

Two important things to consider when implementing 3D audio are the amount of memory the program takes up, and how smooth the transition is between points. The issue of the memory can simply be solved by deleting points from the database. However, once points are deleted, the jump between each point becomes more apparent. We have investigated a solution to this problem using interpolation. Our goal was to implement an existing method of interpolation, and use this to minimize the number of stored impulse responses needed in the CIPIC database to accurately recreate a 3D sound field.

2. METHODS

Throughout testing, we experimented with reduced sets of points to interpolate from (i.e. subsets of the 1250 points of the CIPIC database). Through subjective comparisons we chose a subset consisting of 7 azimuth values (-80, -55, -25, 0, 25, 55, 80) and 13 elevation values (-45, -22.5, 0, 22.5, 45, 67.5, 90, 112.5, 135, 157.5, 180, 202.5, 225). Processing audio using this set did not produce noticeable differences from the full set; using smaller subsets did begin to cause noticeable differences.

2.1 Vector-Based Amplitude Panning

Our method for interpolation was based on work done by G. H. de Sousa & M. Queiroz [3], who outline two different interpolation methods: vector-based amplitude panning (VBAP) and filter pole/zero interpolation. VBAP was chosen due to its mathematical simplicity and for its ease of straightforward implementation. This interpolation technique involves selecting three points on a sphere that form a triangle around the point being interpolated. These three points are chosen by calculating the three nearest points for which an HRIR exists, accomplished by our getNearest3 function, a modified form of getNearestUCDpulse, provided with the CIPIC database. The interpolated impulse is determined by summing scaled versions of the three nearest impulses, using appropriate gains calculated by the following formula:

$$\vec{g} = \vec{p}^{\,t} L_{nmk}^{-1} = [\,p_1 \quad p_2 \quad p_3\,] \begin{bmatrix} l_1^n & l_2^n & l_3^n \\ l_1^m & l_2^m & l_{31}^m \\ l_1^k & l_2^k & l_{31}^k \end{bmatrix}^{-1} \tag{1}$$

g: vector of gains, p: point interpolated,
l_n , l_m , l_k : points interpolated between

These gain coefficients are then used to calculate a weighted sum of the three nearest impulses to find the "interpolated" impulse:

$$\hat{h} = g_1 \cdot h_1 + g_2 \cdot h_2 + g_3 \cdot h_3 \quad (2)$$

h: impulse, g: gain coefficient

Interpolated impulses are calculated separately for both the left and right ears. The mono audio input is filtered using each impulse, and the two channels are combined to produce a stereo audio output.

2.2 MATLAB Scripts for Processing Audio

Much of our work is realized in two MATLAB scripts: TestScript3 and SweepTest. TestScript3 is the final iteration of scripts designed to implement our interpolation in a straightforward manner. This includes steps for loading the HRIR database, inputting a test point, reading the three nearest impulses from the reduced set, interpolating the impulses, filtering example audio using this new impulse, and filtering the audio with the actual nearest impulse (from the unreduced database) for comparison.

SweepTest takes our method one step further by constructing a continuous sweep of audio between two spatial locations. This script uses given start and end points as inputs, and outputs a stereo audio file of the sweep. The mono audio signal is separated into 20ms windows, with an overlap of 10ms. For each segment, the audio is multiplied with a Hann window, the three nearest points to the current spatial location are determined, the impulse is interpolated, and the window is filtered with the impulse. These segments are then recombined using the overlap-add technique. This script is designed so that the method would be the same when ported to a real-time system.

2.3 Correcting for Interaural Time Differences

G. H. de Sousa & M. Queiroz account for interaural time differences, but do not mention any details of how they adjusted for this phenomenon. They say, "We alleviate that problem by synchronizing onset times of adjacent HRIRs that were going to be used in a triangular interpolation, applying gains and mixing, and only then adjusting the overall delay based on a geometric model of the CIPIC recording settings" [3]. We are proposing a method that we think is similar to the one used by the authors in addition to accounting for changes in radial distance from the listener.

Implementing this method requires two steps. First, the impulses must have the onsets aligned. This is done before any audio processing takes place, and is more or less straightforward. Second, after the impulses are interpolated but before they are used to filter the audio, they must be delayed to give the appropriate interaural time difference. This second step is described below.

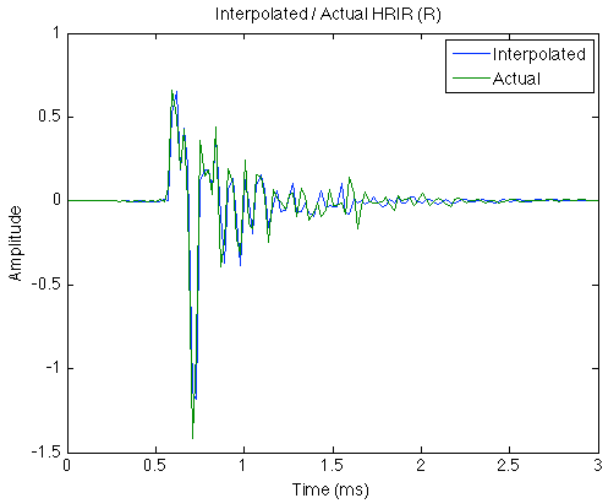

Figure 1: Comparison of actual & interpolated HRIR of the right ear for the point azimuth=20^0, elevation=28.125^0

Let r be the radius of the sphere in the CIPIC experiment with θ and ϕ corresponding to the azimuth and elevation respectively [1]. Also let d be the distance from the center of the head to each ear. Assuming the head is at the center of the sphere, the distance R from a point on the sphere to each ear will be given by (1) and (2) respectively.

$$R_{right} = \sqrt{(d - r\sin\theta\cos\phi)^2 + (r\cos\theta\cos\phi)^2 + (r\sin\phi)^2} \quad (3)$$

$$R_{left} = \sqrt{(-d - r\sin\theta\cos\phi)^2 + (r\cos\theta\cos\phi)^2 + (r\sin\phi)^2} \quad (4)$$

The time (t) it takes sound to reach each ear is found by the following equation, where c is the speed of sound.

$$t = R/c \quad (5)$$

From (3), (4) and (5), we can infer the arrival time of the interpolated point as well as the three measured points. We can then determine the expected interaural time difference Δt. The impulse response will be shifted by $\Delta t * fs$ samples, where fs is the sampling rate. Note that method is merely hypothesizing what the authors could have meant by "adjusting the overall delay based on a geometric model of the CIPIC recording settings" and is at present, unconfirmed [3]. Our tests show that this shift should be no more than 8 samples.

3. RESULTS

3D audio using HRTFs is necessarily a subjective experience as the impulses are recorded using a particular ear and body shape. Impulse responses and frequency responses can be directly compared, but it is not possible to measure how accurately an interpolated impulse a given location, as it is likely to be experienced differently for different listeners.

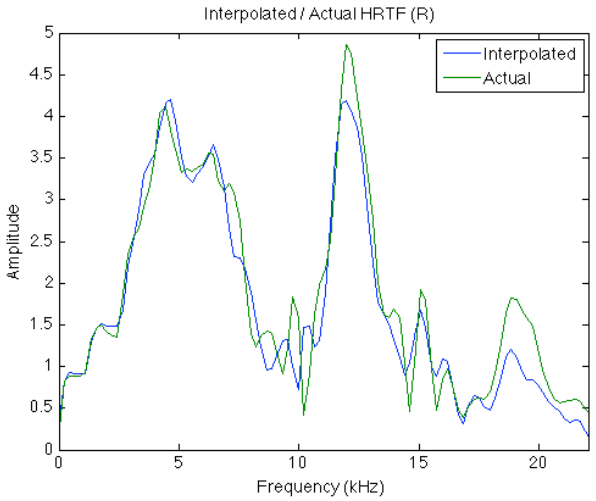

Figure 2: Comparison of actual & interpolated HRTF of the right ear for the point azimuth=20^0, elevation=28.125^0

TestScript3 was run for the point (20, 28,125), which is the location of one of the original impulses from the database, but falls between points of our reduced set. The right-ear actual HRIR and interpolated HRIR are compared in Figure 1, with the HRTFs compared in Figure 2.

Initially, we did not properly account for the interaural time difference. This caused a noticeable difference in perceived location between the audio filtered with actual vs. interpolated HRTF, of approximately 5 degrees. When the impulses were corrected to have the appropriate interaural time difference, there was no perceivable difference in spatial location. The relevant included audio files are example_act.wav (filtered using the actual HRTF), example_int.wav (filtered using the interpolated HRTF), and example_int_corrected.wav (with the correct delays).

SweepTest was used to create 15-second "orbits", where the audio is perceived to be circling the listener's head at various angles. Orbits were created using our reduced set of impulses, as well as the full database set, to interpolate between. There were not perceivable differences between these sweeps. The included orbit wave files are labeled 'all' for those using the full 1250 point, and 'reduced' for those using the reduced set of 91 points.

4. REAL-TIME APPLICATIONS

Our methods from in SweepTest were used to create a plugin for real-time audio processing in RackAFX. While we were able to successfully filter audio in real time for a chosen spatial location, we were unable to progress further due to software malfunctions.

5. CONCLUSIONS

Through our experimentation, we have determined that a reduced set of data points can accurately represent the larger CIPIC database of points when interpolating using our method.

5.1 Further study

One aspect of HRTF interpolation that was not explored in this project was the idea of changing the radial distance of the source from the listener. The CIPIC experimental HRTFs were measured in an anechoic chamber at a radius of 1m from the center of the head [1]. We believe that accurate distances can be simulated by playing the audio through two channels: one with direct sound, and one with reverberation. The reverberation will simulate the type of room the listener is in, while the direct sound will be an unmodified version of the audio.

The reverberation channel would be set at a constant level. The gain of the direct sound will then be adjusted proportionally to the distance from the listener. At small radii, the direct sound will be significantly higher than the reverberation, and the source will appear closer to the listener. At large radii, the reverberated channel will be higher than the direct sound, and the source will appear further away from the listener. If the listener is outside, where there is almost no reverberation, the radius of the sound source can be accurately adjusted by changing the volume of the audio. If this idea is successful, no additional HRTFs need be measured, and the entire 3D sound field can be accurately replicated using only 91 measured points.

Another potential method for reducing the necessary points would be to have only one data set of impulses for the set of points in space, instead of having two impulses, one for each ear. If the system is assumed to be symmetric (which often is not the case due to differences in one's left and right ears), then each point could be matched with the corresponding point having the opposite azimuth. Investigation could be done as to whether or not such a method could be used to reasonably reconstruct filters for both ears using one data set.

6. REFERENCES

[1] Algazi, V. R., R. O. Duda, D. M. Thompson, & C. Avendano, "The CIPIC HRTF database", *Applications of Signal Processing to Audio and Acoustics, 2001 IEEE Workshop on the Applications of Signal Processing to Audio and Electroacoustics* (pp. 99-102), IEEE, (2001)

[2] CIPIC Database - Copyright (c) 2001 The Regents of the University of California. All Rights Reserved

[3] de Sousa, G. H., & M. Queiroz, "Two approaches for HRTF interpolation", *The 12th Brazilian Symposium on Computer Music (SBCM 2009),* (2009, September).

[4] Doukhan, D., & A. C. Sédès, "A Binaural Synthesis External for Pure Data". *PD Convention*, (2009)