Audio Code Proposal - Fall 2012

Kris Kooi

September 7, 2012

Current algorithm

Due to limitations of Dalvik and mobile devices in terms of OS support, CPU speed, and memory, the app does not simulate a full 3-D environment. Instead, it spacializes sound along the horizontal plane of the ears and to the front of the user. Spacialization is achieved by simulating interaural time delay (ITD) and interaural level difference (ILD). In order to simplify the environment, the following assumptions were made:

- Radius of a human head: 8.5 cm
- Ears are points lying on the opposite ends of the sphere (no outer ears)
- Room temperature is assumed to be $20°C$
- Speed of sound is 343.42 m/s
- There is no reverberation in the location
- The ear perceives all sound in the same fashion regardless of pitch

Interaural Time Delay is the difference in time it takes for the two ears to hear the sound. With the angle θ is defined as being 0 directly in front of the face, $\frac{\pi}{2}$ when on the line formed by the center of the head and the right ear, and -$\frac{\pi}{2}$ when on the line formed by the center of the head and the left ear, ITD is calculated with the formula :

$$InterauralTimeDelay(s) = \frac{radius_{head}}{speed_{sound}} \times (\theta + sin\theta) \tag{1}$$

Negative ITD indicates that the left ear hears the sound first. θ is expressed in radians.

Interaural Level Difference is the difference in volume perceived by the two ears. According to the distance law, the soud pressure (volume) emitted by a source is inversely proportoinal to its distance from the listener.

$$\Delta_{distance} = ITD(s) \times speed_{sound} \tag{2}$$

$$distance_{left} = distance_{center} + \frac{\Delta_{distance}}{2} \tag{3}$$

$$distance_{right} = distance_{center} - \frac{\Delta_{distance}}{2} \tag{4}$$

where $distance_{center}$ is the distance from the sound source to the center of the head. Volume is set for each ear accordingly.

Problems with Current Implementation

The current implementation suffers when $\theta = \frac{\pi}{4}$ or $\theta = \frac{-\pi}{4}$. In this case, the ITD for a sound located in front of the head and one located behind the head are identical. This case is called the "cone of confusion," so named by the cone created when this case is projected into a 3D sound environment. In natural environments, the cone of confusion is resolved with cues from the outer ear, acoustic variances in the environment, or movement of the head. None of these are available given our current set of assumptions about our virtual sound environment.

Our current implementation also does not account for sounds of different frequencies. Low pitched (¡1000 Hz) sounds are primarily localized by ITD, whereas higher pitched sounds are primarily localized through ILD. As long as our application's main use is transmitting the human voice, this is no serious detriment. As there is no easy way to determine the frequency of the signal on Android, it is unlikely that we will be able to differentiate between low and high pitched sounds.

Though there is nothing actually wrong with the ILD equation our current implementation uses, Birchfield and Gangishetty have proposed a different one that is worth consideration. They showed experimentally that their equation for calculating ILD generated sounds that their test subjects were able to place quite accurately in a virtual sound space even without IDL.

Proposed Changes

We will provisionally replace the ILD algorithm from our current implementation with the algorithm proposed by Birchfield and Gangishetty:

$$\begin{bmatrix} x & y & 1 \end{bmatrix} \begin{bmatrix} c_e & 0 & -c_x \\ 0 & c_e & -c_y \\ -c_x & -c_y & c \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = 0 \tag{5}$$

where E_1 and E_2 are the energies of the signals received by the two ears and
$c_e = E_1 - E_2$
$c_x = E_1 x_1 - E_2 x_2$
$c_y = E_1 y_1 - E_2 y_2$
$c = E_1(x_1^2 + y_1^2) - E_2(x_2^2 + y_2^2)$.

This algorithm can be further refined into two more specific cases. When the energies of the two signals are not equal, which is when $\theta \neq 0$, then the equation reduces to

$$(x - \frac{c_x}{c_e}) + (y - \frac{c_y}{c_e}) = \frac{E_1 E_2 d_{12}^2}{c_e^2} \tag{6}$$

where $d_{12} = (x_1 - x_2)^2 + (y_1 - y_2)^2$ is the squared distance between the two microphones. In this case, the sound source lies on a circle that surrounds the ear. ITD allows the user's ears to determine where exactly on the circle the sound originated.

When the energies of each signal are equal, which is to say that $d_1 = d_2$ and $\theta = 0$, then the equation reduces to

$$2c_x x + 2c_y y = c \tag{7}$$

which is the equation of the perpendicular bisector of the line joining the two ears. In this case, ITD will not be able to help the user discerne where on the line the sound originated, so for this case, the current ILD equation will be used.

Implementation and Timeline

As in the current implementation, we will be pre-calculating ITD and ILD for 121 regions of our conference room, barring any drastic changes to the UI. We will focus our first one or two cycles on implementing the new ILD algorithm and ensuring its stability. From there, we will explore optimizations to improve the speed at which the application is able to move speakers around the virtual sound space. We will pay particular interest to Android's OpenSL capabilities, as we may be able to save some of the overhead of Java's object-orientation by manipulating the audio streams in native code.

As the algorithm we will be implementing is necessarily somewhat experimental, the timeline for the full semester cannot be fully predicted. I expect that in roughly one month's time (early October) we will want to submit our implementation for user testing to determine the success of our new ILD algorithm.

Selected Sources

Garton,B. Two New Approaches to the Simulation of Acoustic Spaces.

"Head-related Transfer Function." Wikipedia. Wikimedia Foundation, 14 June 2012. Web. 15 June 2012. ¡http://en.wikipedia.org/wiki/Head-related_transfer_function¿.

"Sound Localisation." Wikipedia. Wikimedia Foundation, 06 Sept. 2012. Web. 13 June 2012. ¡http://en.wikipedia.org/wiki/Sound_localisation¿.

"Binaural Recording." Wikipedia. Wikimedia Foundation, 14 June 2012. Web. 13 June 2012. ¡http://en.wikipedia.org/wiki/Binaural_recording¿.

Malham,D and Myatt,A (1995) 3-D Sound Spatialisation using Ambisonic Techniques. Computer Music Journal v.19,n.4,pgs.58-70

Pulkki,V. (2000) Generic Panning Tools for MAX/MSP. Proceedings of the International Computer Music Conference

Sun,H., et al. (2012) Optimal Higher order Ambisonics Encoding With Predefined Constraints. Transactions on Audio,Speech, And Language Processing, Vol.20,n.3,pgs.742-754

Ahrens,J and Spors,S. (2008) Analytical Driving Functions For Higher Order Ambisonics. IEEE international conference on acoustics, speech, and signal processing.

Jeub,M., et al (2010) Model-Based Dereverberation Preserving Binaural Cues Transactions on Audio,Speech, And Language Processing. Vol.18,n.7, pgs 1732-1745.

Cheng, C. and Wakefield, G. (2001) Introduction to Head-Related Transfer Functions (HRTFs): Representations of HRTPs in Time, Frequency, and Space. Journal of the Audio Engineering Society, Vol 49 No. 4.

Birchfield, S. and Gangishetty, R. (2005) Acoustic Localization by Interaural Level Difference. IEEE International Conference on Acoustices, Speech, and Signal Processing (ICASSP). Philadelphia, Pennsylvania, March 2005.