

A HANDS-ON APPROACH TO BINAURAL SPATIAL AUDIO EDUCATION

Kenneth John Faller II and Christopher Nguyen
Computer Engineering Program
California State University, Fullerton

Armando Barreto
Electrical & Computer Engineering Department
Florida International University

Abstract

This paper describes a Matlab program designed to facilitate teaching techniques for generating binaural spatial audio using Head-Related Transfer Functions (HRTFs). The program has an intuitive interface with several interactive visual and auditory examples. Additionally, an embedded program is described which allows for real-time processing of HRTFs using the low-cost Texas Instruments (TI) C5515 eZdsp USB Stick Development Tool. Code for both the Matlab and embedded programs, and an example Matlab project are shared with interested educators at a designated web site.

Introduction

Immersive environments, such as virtual reality and computer gaming, require three-dimensional (3D) visual and audio capabilities to generate a highly realistic experience for the user. Considering the growth in these technical areas [1] and the impact of spatial audio on them, it is important to provide students in relevant technical fields (i.e., computer engineering, computer science, etc.) with a basic understanding of the techniques and concepts of generating 3D binaural spatial audio using computers.

The purpose of this paper is to provide interested educators with freely available and prepackaged embedded programs and Matlab software that can be used in their lectures. The Matlab software is an updated version of the implementation described in [2]. The paper is organized to first provide the reader with a basic understanding of spatial hearing and DSP techniques for generating binaural spatial audio using a computer. Then the Matlab and

embedded programs are described. Finally, an example Matlab project that can be incorporated into existing courses (i.e., Digital Signal Processing (DSP) related courses) is described.

Spatial Hearing

A sound wave is transformed as it travels from the source to the entrance of the ear canal. It is theorized that these transformations serve as cues for human sound localization. The transformations can be accurately measured with sophisticated equipment (e.g., the AuSIM HeadZap system [3]) and used to create special digital filters called Head-Related Transfer Functions (HRTFs) which can then be used to spatialize binaural audio. The following subsections will give the reader a basic understanding of spatial hearing. For more in-depth reading on the topic of spatial hearing, refer to [4].

Spherical Coordinate System

For this paper a spherical coordinate system, specified by azimuth (θ), elevation (Φ) and distance (d), is used to specify locations in 3D space (see Figure 1). The azimuth varies along the horizontal plane with 0° representing the position in front of the subject. As the position is moved from the center towards the right, the azimuth increases from 0° to $+90^\circ$. The azimuth signs are reversed in the left hemisphere, moving from 0° to -90° . The positions directly perpendicular to the left and right ears are represented by -90° and 90° , respectively. For this paper, the elevation is fixed to 0° which represents an intersection with the horizontal plane (Figure 2). Furthermore, the available azimuths are from -90° and 90° at intervals of 10° (see Figure 3).

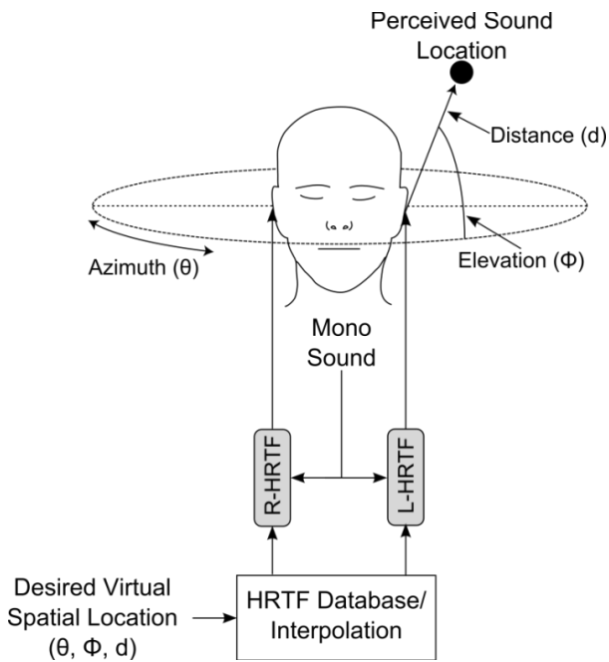


Figure 1: HRTF modeling of sound transformation from a source to each of the listener's eardrums.

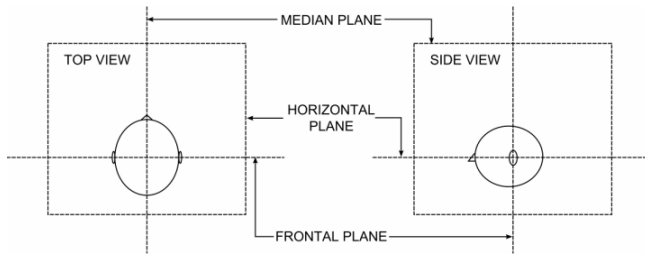


Figure 2: The median, horizontal, and frontal planes (image from [5]).

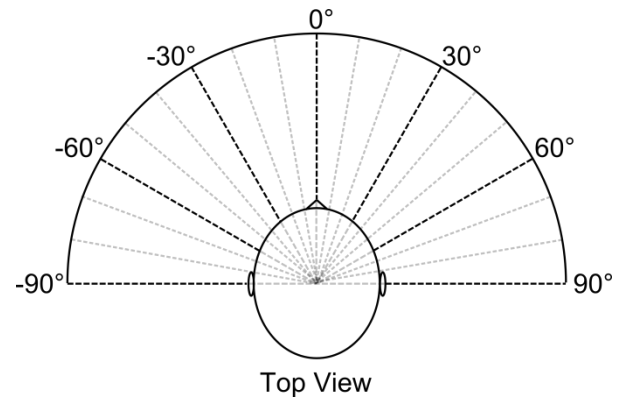


Figure 3: Azimuth locations.

Azimuthal Cues

The Duplex theory describes two binaural cues, called Interaural Time Difference (ITD) and Interaural Intensity Difference (IID), which can be used to achieve azimuthal spatialization [6]. ITDs are the difference in arrival times and IIDs are the difference in intensities of a sound between the left and right ears (see Figure 4). Generally for IIDs, the sound will be perceived at a larger intensity for the ipsilateral ear (i.e., ear closest to the sound source). Similarly for ITDs, the sound will arrive earlier for the ipsilateral ear.

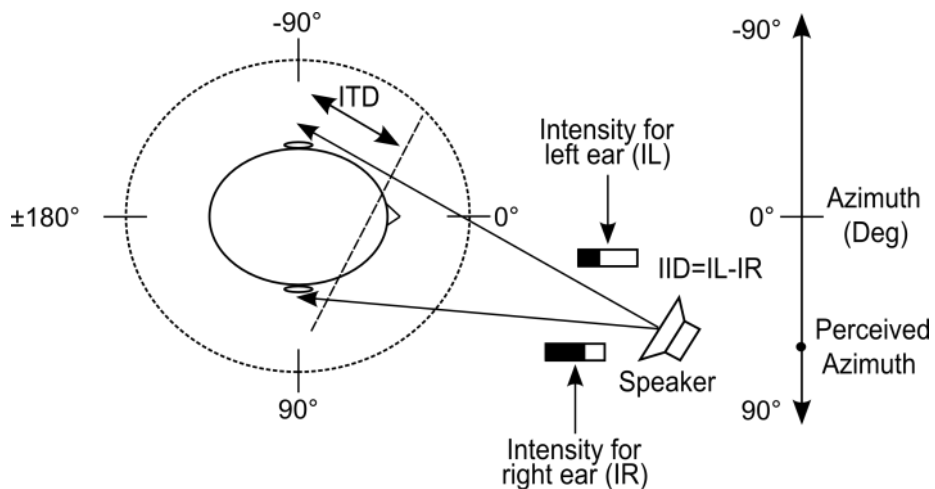


Figure 4: Estimate of azimuth of a sound source using ITD and IID (image from [5]).

However, ITDs and IIDs have some localization limitations. For example, at certain frequencies the ITD cue becomes ambiguous because the wavelength of the sound and the diameter of the head are comparable (e.g., 1.5 kHz) [7]. Another problem is the reversal errors (i.e., front-to-back confusions) which are caused by symmetric paths along the horizontal or median planes that result in identical ITDs and IIDs (see Figure 5). Nonetheless, the Duplex Theory gives a simple and intuitive explanation of how humans localize azimuthally and has resulted in many spherical head models for audio spatialization [8-10].

Cone-of-Confusion

As mentioned in the previous section, sound source locations that are symmetric with respect

to an imaginary axis passing through both ears (interaural axis) will have virtually identical ITDs and IIDs which can lead to confusion about the true location of the sound source. The locations around the listener which have the same ITDs and IIDs are said to lay in a “cone of confusion”[11]. To further clarify, Figure 6 shows four locations which a listener could confuse. For example, the locations of A and C are situated such that they have virtually identical ITDs and IIDs which could result in confusion (up vs. down) between these two source locations. The same situation could occur for locations B and D but, in that case, the confusion occurs between a point in the front hemisphere (B) and a symmetric point in the back hemisphere (D).

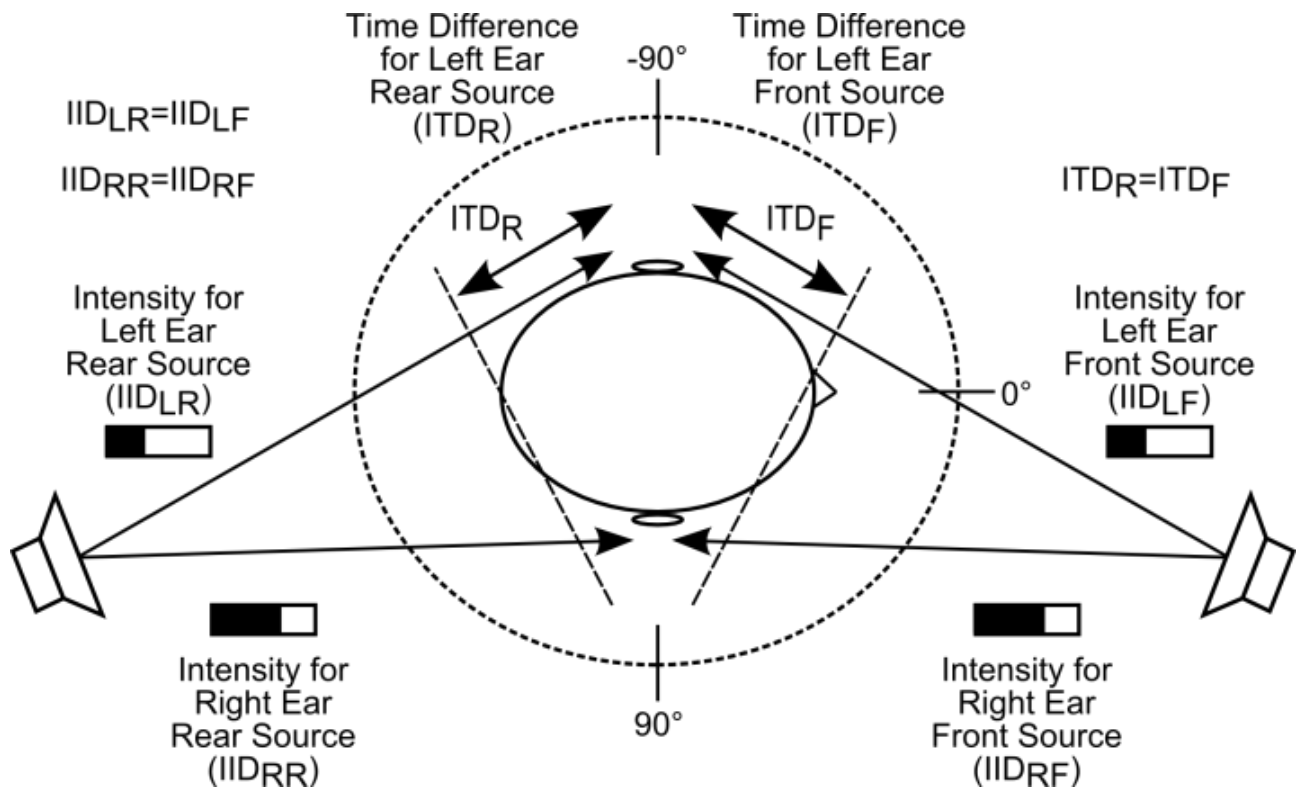


Figure 5: Symmetric sound sources have identical IIDs and ITDs which results in front-to-back confusion.

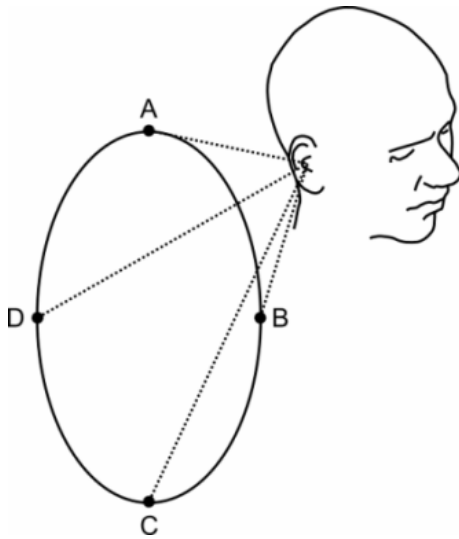


Figure 6: Cone of confusion – Sounds from position A could be mistaken as originating from C. Similar confusion may exist between B and D, and other symmetrical positions on the cone,(Image from [5]).

Head-Related Transfer Functions (HRTFs)

Head-Related Transfer Functions (HRTFs) (Figure 7, top) are Linear Time-Invariant (LTI) transfer functions that characterize the specific transformation that sounds experience from a source anywhere around a subject to his/her ear canals. This implies that a sound will be processed by a pair of HRTFs, one for the left channel and one for the right channel. The pair of HRTFs corresponds to a particular 3D location (see previous section). As is the case for all LTI dynamical systems, the transformations can also be characterized by the impulse response signals associated with the transformations which, in this case, are called Head-Related Impulse Responses (HRIRs) (Figure 7, bottom).

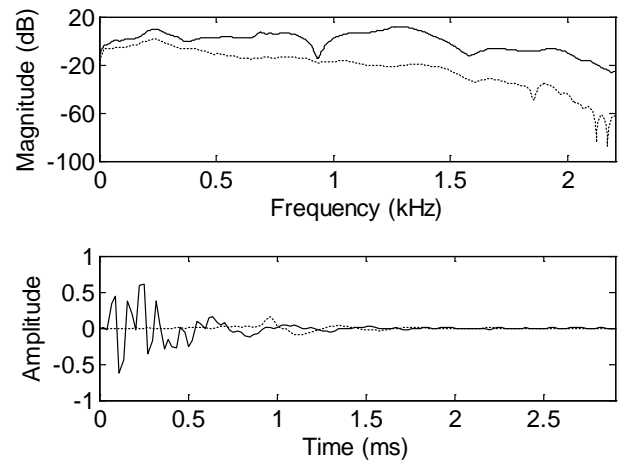


Figure 7: HRTF magnitude responses (top) and HRIRs (bottom) of the right (solid) and left (dotted) channels with the sound source at azimuth=90° and elevation=0° (image from [12]).

The HRTFs can be used to synthesize sound by filtering a digitized monaural audio input with the appropriate pair of HRTFs to produce a stereo (left and right channel) output (Figure 1). When the resulting output is played to a listener through headphones, he/she will perceive the sound as if it emanated from the desired location in 3D space.

The filtering process is achieved by convolving the digitized input sound with the appropriate pair of HRIRs. HRIRs are generally modeled as Finite Impulse Response (FIR) filters. As a result, the difference equation shown in eq. (1) gives the relationship between the digitized input (x) and the filtered output (y) where n is the discrete time index, “*” is the convolution operator, and b_M is the M^{th} coefficient of the HRIR (h).

$$\begin{aligned}
 y[n] &= b_0x[n] + b_1x[n - 1] + b_2x[n - 2] \\
 &\quad + \dots + b_Mx[n - M] \\
 &= \sum_{m=0}^M b_mx[n - m] \\
 &= \sum_{m=-\infty}^{\infty} h[m]x[n - m] \\
 &\triangleq (h * x)[n]
 \end{aligned} \tag{1}$$

Alternately, the input-output description of this filtering operation in the z-transform domain is a rational transfer function (see eq.(2)). For a more thorough explanation of using digital filters for audio applications refer to [13].

$$Y(z) = (b_0 + b_1z^{-1} + b_2z^{-2} + \dots + b_Mz^{-M})X(z) \quad (2)$$

HRTF Database

Measurement of HRTFs requires expensive and cumbersome equipment. Fortunately, several researchers have created freely available HRTF databases such as the CIPIC database [14] and the MIT HRTF database [12]. In practice, these systems replace a theoretical impulse input with Golay Codes [15] or Maximum Length Sequences (MLS) [12] and record the response from microphones placed at the ear canal of the subject.

For this paper, the HRTFs obtained from measuring a KEMAR mannequin will be used (i.e., MIT’s HRTF database [12]). The sampling rate used for this database is 44.1 kHz which is also the sampling rate used for the Matlab simulation and project (see next section). For the embedded program, the HRTFs need to be resampled[16] to a 48 kHz sampling rate which is the rate used by the development tool (see next section). The original HRIR length is 128 samples. After the resampling, the HRIR length is extended to 140 samples.

Supplemental Materials

The following subsections describe the supplemental materials that accompany this paper which include the Matlab based HRTF teaching software, a real-time embedded program, and an example HRTF project. Also included are brief explanations of how to setup and use the supplemental materials. All of these materials are included in a single ZIP file at:

http://ecs.fullerton.edu/~jfaller/supp_mats.zip

Matlab HRTF Teaching Software

The Matlab HRTF teaching software is an updated version of the software described in [2]. However, the “Filter Demo” mode is removed and several features, such as the capability to select the audio file, audio looping (i.e., audio file plays on a loop), and play and stop buttons, are added (see Figure 8). All of the other previous capabilities are still available.

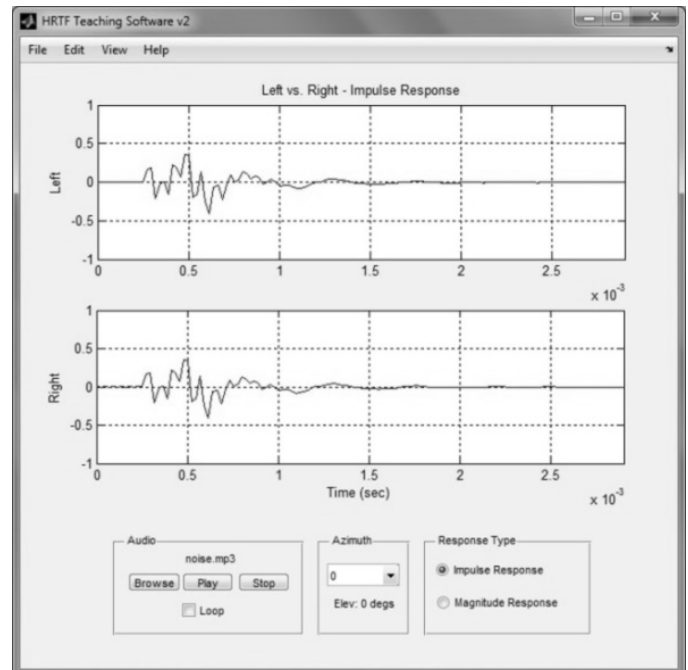


Figure 8: Screenshot of the GUI for the HRTF teaching software.

The user can choose a desired azimuth from a drop down list in the section labeled “Azimuth” where the default azimuth angle is 0°. In the section labeled “Response Type,” the user can choose between magnitude or impulse response plot types where the impulse response is the default. The Matlab functions used for plotting the magnitude and impulse responses are shown in Figures 10 and 11, respectively. Finally, the user can also select an audio file using the “Browse” button. The acceptable audio file types are WAV or MP3 files. The default audio file is the provided “noise.mp3” file which is 1 second of band-limited white noise. The audio file must be sampled at 44.1 kHz which is the

sampling rate of the HRTFs. Hence, if the selected audio file is of a different sampling rate, it will be resampled[17]. Once the appropriate selections are made, the “Play” button can be used to play the filtered audio. This invokes the playBtn_Callback function which subsequently calls the playAudio function (see Figure 9). The “Loop” checkbox, when checked, will loop the audio output until the “Stop” button is pressed.

```

Function playAudio(handles)
Global filename pathName player;

[HL, HR, Fs]=getHrtfs(handles);

[audio, Fs_Audio]=
audioread(cat(2,pathName,fileName));

if(Fs ~=Fs_Audio)
msgBoxH = msgbox('One moment,
    resampling...', 'Resampling', 'modal');
audio= resample(audio,Fs,Fs_Audio);
close(msgBoxH);
end

info = audioinfo(cat(2,pathName,fileName));
if(info.NumChannels> 1)
audio = audio(:,1);
end

% Filter
[FHL, FHR]=hrtfFilter(HL,HR,audio);

% Audio Out
audioOut=[FHL, FHR];
player = audioplayer(audioOut, Fs);
playblocking(player);

```

Figure 9: Function called when the “Play” button is pressed (playAudio function in main.m).

```

Function
plotMagResp(HL,HR,axesLeft,axesRight,Fs)
% function
plotMagResp(HL,HR,axesLeft,axesRight,Fs)
%
% Function that plots the magnitude responses
for given
% left and right HRTFs
%
% Syntax:
% [FHL, FHR] =
hrtfFilter(hrtfL,hrtfR,audioInput);
%
plotMagResp(FHL,FHR,axesLeft,axesRight,44100)
;
%

```

```

axes(axesLeft);
cla;
[HL,FL]=freqz(HL,1,1000,Fs);
plot(FL,mag2db(abs(HL)));
xlim([0 Fs/2]);
ylim([-60 20]);
grid on;
title(axesLeft,'Left vs. Right - Magnitude
    Response');
ylabel(axesLeft,'Left Magnitude (dB)');

axes(axesRight);
cla;
[HR,FR]=freqz(HR,1,1000,Fs);
plot(FR,mag2db(abs(HR)));
xlim([0 Fs/2]);
ylim([-60 20]);
grid on;
ylabel(axesRight,'Right Magnitude (dB)');
xlabel(axesRight,'Frequency (Hz)');

```

Figure10: Function that plots the magnitude response (plotMagResp.m).

```

function
plotImpulseResp(HL,HR,axesLeft,axesRight,Fs)
% function
plotImpulseResp(HL,HR,axesLeft,axesRight,Fs)
%
% Function that plots the impulse responses
for given
% left and right HRTFs
%
% Syntax:
% [FHL, FHR] =
hrtfFilter(hrtfL,hrtfR,audioInput);
%
plotImpulseResp(FHL,FHR,axesLeft,axesRight,44
100);
%
axes(axesLeft);
cla;
t =linspace(0,length(HL)/Fs,length(HL));
plot(t,HL);
grid on;
box on;
xlim([0 length(HL)/Fs]);
ylim([-1 1]);
title('Left vs. Right - Impulse Response');
ylabel('Left');

axes(axesRight);
cla;
t =linspace(0,length(HR)/Fs,length(HR));
plot(t,HR);
grid on;
box on;
xlim([0 length(HR)/Fs]);
ylim([-1 1]);
ylabel('Right');
xlabel('Time (sec)');

```

Figure 11: Function that plots the impulse response (plotImpulseResp.m).

Embedded Software

The development kit used is the Texas Instruments (TI)C5515 eZdsp USB stick development tool (approximately \$79) [18]. This kit contains a TI C5515 low power 16-bit fixed point DSP with 240 MIPS performance, 320KB on-chip memory, and a TI AIC3204 32-bit programmable low-power stereo audio COder-DECoder (CODEC). The kit also has several external peripherals (see Figure 12) including two 1/8" audio connectors (solid box), pushbuttons (dashed box), and a LCD screen (arrow).

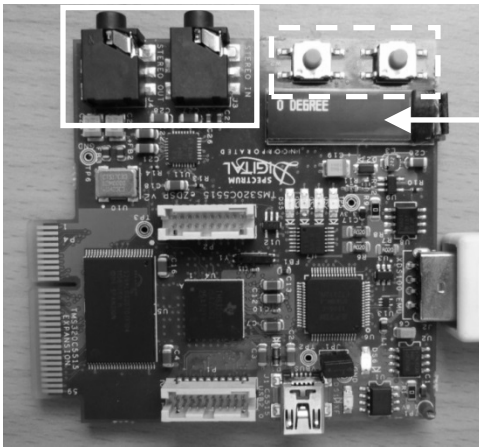


Figure 12: TI C5515 eZdsp USB stick development tool.

The user can feed a stereo signal into the input 1/8" audio connector. However, considering only a monaural input is required (see Figure 1), only the left channel is used during the filtering process. The user can select the azimuth angle by pressing the pushbuttons while looking at the current azimuth angle displayed on the LCD display. Initially, the azimuth is set to 0°. The output of the filtering process is then sent out through the output 1/8" audio connector. Output from this system requires stereo headphones for proper spatial localization of the sound source.

All of the embedded software is written in C or C-callable assembly programming languages using the Code Composer version 4.1.3 IDE. The code is based on TI's C5515 audio filter demo [19] which contains code for initializing and configuring several of the peripherals on the

kit such as the Direct Memory Access (DMA) controller, I²C interface, the audio CODEC, etc. Additionally, several DSP related functions are also provided such as the FIR filter function ("fir.asm").

HRTF Coefficients

As mentioned, these HRTFs were measured on a KEMAR mannequin which has symmetric anatomical features. As a result, it is assumed that the HRTFs are symmetric about the median plane (see Figure 2). Hence, only coefficients for HRTFs corresponding to locations from 0° to 90° at intervals of 10°, for both the left and right ears, are stored in the on-chip memory (i.e., a total of 20 sets of 140 coefficients). All the HRTF filter coefficients are in the "ref_data_bypass.c" file and are stored as 16-bit integers.

Ping-Pong Buffer

The Ping-Pong mode (a.k.a., page flipping) for DMA transfer can be used to continuously process incoming audio data without sample loss. As shown in Figure 13, the DMA starts filling up the Ping buffer 1 first. Once the Ping buffer 1 is full, a DMA interrupt is generated to the CPU which indicates that the data in the Ping buffer 1 can be processed. The CPU can then start processing the samples in the Ping buffer 1 while the DMA is transferring the data to the Pong buffer 1. When the CPU is finished processing the data, the data is written to the Ping buffer 2. The DMA will then send the data to the audioCODEC. The DMA will switch between the Ping and Pong buffers for the write and read process.

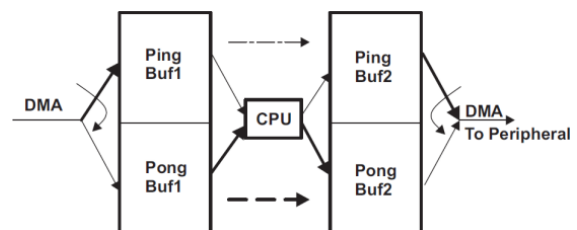


Figure 13: Ping-Pong mode for DMA data transfer (from [20]).

Example Project

Also provided with the supplemental materials is an example project that includes Matlab code, a lecture, and lab instructions. The purpose of this project is to allow for easy incorporation into an existing DSP-based course. A Matlab GUI is provided which displays a top view of the front portion of a human head (Figure 14). Using the mouse, the user can select the position of the source which will be at 0° elevation and azimuth angles between -90° and 90° .

The GUI will track the mouse pointer and it will display the xy-coordinates in the Matlab command window as shown in Figure 14. This GUI is programmed to call the “processHrtf” function which the students will program. The current xy-coordinates of the crosshair mouse pointer shown in Figure 14 will be passed to the function when the left mouse button is pressed. With these coordinates, the students are to do the following in the “processHrtf” function (“processHrtf.m” file):

1. Convert the xy-coordinates of the mouse pointer to the azimuth angle in degrees.
2. Determine the appropriate pair of HRTFs closest to the selected azimuth angle. For example, a selected azimuth angle of 12° would use the HRTFs for azimuth 10° . The angle is rounded up to the larger azimuth angle if it is in between discrete angles. For example, a selected azimuth angle of 15° would use the HRTFs for azimuth 20° . Only an elevation of 0° and azimuths from -90° to 90° at intervals of 10° are to be simulated. The selected azimuth angle is then to be displayed in the Matlab command window as shown in Figure 14.
3. The xy-coordinates of the mouse pointer are to be used to calculate the distance from the center of the head. The distance is then to be displayed in the Matlab command window as shown in Figure 14.
4. Use the distance to calculate the appropriate

gain.

5. Generate one second of white noise (see the Matlab “randn” command) and then filter it using the selected HRTFs.
6. Scale the filtered output using the calculated gain factor.
7. Finally, play the resulting filtered stereo sound using the Matlab “sound” command.

A lecture is also provided which can be used to explain to the students how to do the various conversions and calculations needed to complete the project (see the “Project_Lecture.pptx” file under the “Project” folder of the supplemental materials).

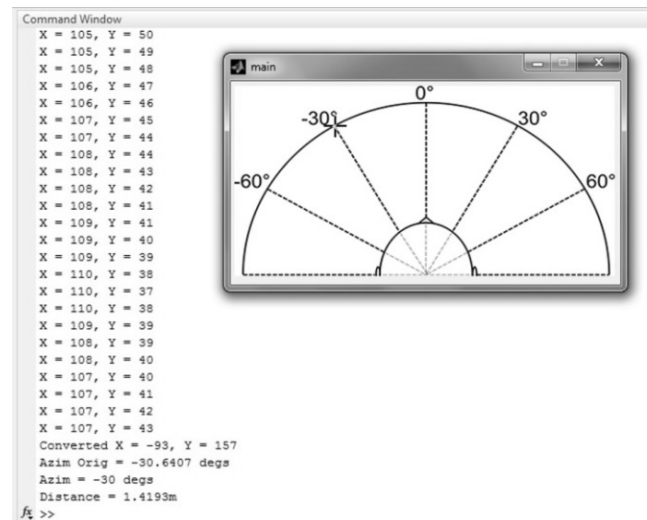


Figure 14: Matlab GUI and command window for the project.

Results and Discussion

The use of the Matlab teaching software, the embedded program on the TIC5515 eZdsp USB Stick Development Tool, and the Matlab projects were successfully integrated into some of the first author’s undergraduate courses (i.e., a course on wearable computers and augmented reality, and an independent study). Overall, students reported that they were able to accurately localize the virtual sound sources. However, some students did report front-to-back confusions. As discussed, this is a common

phenomenon resulting from the symmetry of the azimuthal cues (i.e., ITD and IID) about the median plane (see Figure 2). These errors are further exacerbated by the use of non-individualized HRTFs [21]. To achieve the best accuracy, the HRTFs should be individually measured for the subject. Unfortunately, the equipment to measure HRTFs is cumbersome and expensive and, therefore, not available to the general public.

Conclusion

Education on binaural spatial audio has been largely neglected. More specifically, demonstrating the use of HRTFs for binaural spatial audio on a real-time system is not traditionally taught in DSP courses. Additionally, DSP textbooks and education literature do not generally include topics on binaural spatial audio. This paper demonstrates spatial audio concepts through a Matlab based teaching tool and an embedded system that runs on a low cost TI C5515 eZdsp USB stick development tool. Additionally, a sample lecture and a project are provided. Interested educators and students can obtain all the supplementary materials at the web site given in the paper. The materials are provided to facilitate the incorporation of the topic into existing DSP courses and they can be used to teach binaural spatial audio concepts and more fundamental DSP concepts such as: transfer functions, impulse response, convolution, frequency response, etc.

References

1. Entertainment Software Association (ESA), "Essential Facts About the Computer and Video Game Industry," Washington, DC, 2013.
2. K. Faller II and A. Barreto, "Simulation and Real-Time Implementation for Teaching 3D Sound," ASEE Computers in Education Journal, vol. XVI, no. 2, pp. 36–43, 2006.
3. AuSIM Inc., "Corporate Website," 2012. [Online]. Available: <http://www.ausim3d.com/>. [Accessed: 08-Mar-2014].
4. J. Blauert, *Spatial Hearing: The Psychophysics of Human Sound Source Localization*, 2nd ed. Cambridge, MA: MIT Press, 1997.
5. K. Faller II, "Reduced-Parameter Model of Head-Related Transfer Functions for Synthesized Spatial Audio," Florida International University, 2009.
6. Lord Rayleigh, "On Our Perception of the Direction of a Source of Sound," *Proceedings of the Musical Association*, vol. 2, no. 341, pp. 75–84, 1875.
7. C. I. Cheng and G. H. Wakefield, "Introduction to Head-Related Transfer Functions (HRTFs): Representations of HRTFs in Time, Frequency, and Space," *Journal of the Audio Engineering Society*, vol. 49, no. 4, pp. 231–249, 2001.
8. R. O. Duda and W. L. Martens, "Range Dependence of the Response of a Spherical Head Model," *Journal of the Acoustical Society of America*, vol. 104, pp. 3048–3058, 1998.
9. B. Shinn-Cunningham, S. Santarelli, and N. Kopco, "Tori of Confusion: Binaural Cues for Sources Within Reach of a Listener," *Journal of the Acoustical Society of America*, vol. 107, pp. 1627–163, 1999.
10. V. Algazi, C. Avendano, and R. Duda, "Estimation of a Spherical-Head Model from Anthropometry," *Journal of the Audio Engineering Society*, vol. 49, pp. 472–479, 2001.
11. J. Blauert, "Sound Localization in the Median Plane," *Acta Acustica united with Acustica*, vol. 22, no. 4, pp. 205–213, 1969.
12. B. Gardner and K. Martin, "HRTF Measurements of a KEMAR Dummy-

- Head Microphone,” Massachusetts Institute of Technology (MIT) Media Laboratory Vision and Modeling Group. [Online]. Available: <http://sound.media.mit.edu/resources/KEMAR.html>. [Accessed: 08-Mar-2014].
13. T. B. Welch, C. H. G. Wright, and M. G. Morrow, *Real-Time Digital Signal Processing from MATLAB to C with the TMS320C6x DSPs*, 1st ed. Boca Raton, FL: CRC Press, 2006, p. 363.
 14. V. Algazi et al., “The Cipic HRTF Database,” IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA), 2001. [Online]. Available: <http://interface.cipic.ucdavis.edu/>. [Accessed: 08-Mar-2014].
 15. S. Foster, “Impulse Response Measurement Using Golay Codes,” IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), vol. 11, pp. 929–932, 1986.
 16. J. O. Smith, “Digital Audio Resampling Home Page,” Center for Computer Research in Music and Acoustics (CCRMA), Stanford University, 2002. [Online]. Available: <http://www-ccrma.stanford.edu/~jos/resample/>. [Accessed: 09-Jun-2014].
 17. J. O. Smith, “Digital Audio Resampling Home Page,” Center for Computer Research in Music and Acoustics (CCRMA), Stanford University, 2002. [Online]. Available: <https://ccrma.stanford.edu/~jos/resample/>. [Accessed: 13-Jun-2014].
 18. Texas Instruments (TI), “C5515 eZdsp USB Stick Development Tool,” 2010. [Online]. Available: http://www.ti.com/tool/tmdx5515_ezdsp. [Accessed: 05-Jun-2014].
 19. Texas Instruments (TI), “C5515 eZdsp Audio Filter Demo,” C5000-based eZdsp USB Stick Development Tool, 2010. [Online]. Available: <https://code.google.com/p/c5505-ezdsp/>. [Accessed: 06-Jun-2014].
 20. Texas Instruments (TI), “TMS320C5515/14/05/04 DSP Direct Memory Access (DMA) Controller,” User Guide, 2010. [Online]. Available: <http://www.ti.com/lit/ug/spruft2a/spruft2a.pdf>. [Accessed: 12-Jun-2014].
 21. E. Wenzel et al., “Localization Using Nonindividualized Head-Related Transfer-Functions,” *Journal of the Acoustical Society of America (JASA)*, vol. 94, pp. 111–123, 1993.

Biographical Information

Kenneth John Faller II received his B.S. and M.S. degrees in Computer Engineering from Florida International University (FIU) in Miami, Florida, in 2003 and 2006 respectively. He received his Ph.D. in Electrical Engineering from FIU in 2009. Prior to joining California State University, Fullerton in 2011, he was a NASA postdoctoral fellow. His research interests are HRTF modeling and individualization, spatial audio signal processing, and vocal credibility assessment.

Christopher Nguyen is an undergraduate Computer Engineering student at California State University, Fullerton (CSUF) in Fullerton, CA. His anticipated graduation date is spring 2017. He is also pursuing a minor in Computer Science.

Armando Barreto received his undergraduate Electrical Engineering degree from UNAM (Mexico City) in 1987. He obtained an M.S. in Electrical Engineering from Florida International University (FIU) in Miami, Florida, in 1989, and the Ph.D. in Electrical Engineering from The University of Florida, Gainesville, in 1993. His doctoral research focused on Digital Signal Processing (DSP), with application to electrocorticogram (ECoG) signals. Since 1994, he has been a faculty member of the Electrical and Computer Engineering Department at FIU and the director of FIU’s DSP Laboratory. He is a senior member of IEEE and ACM.