Audio Engineering Society

Convention e-Brief 129

Presented at the International Conference on
Immersive and Interactive Audio
2019 March 27–29, York, UK

Multichannel Audio Implementation for Virtual Reality

Sungsoo Kim, Sripathi Sridhar

New York University, 35 W. 4th St, New York, NY 10012

Correspondence should be addressed to Sungsoo Kim (sk6740@nyu.edu)

ABSTRACT

This paper describes a system that implements audio over a multichannel loudspeaker system for virtual reality (VR) applications. Real-time tracking data such as distances between the user and loudspeakers and head rotation angle are used to modify the output of a multichannel loudspeaker configuration in terms of panning, delay and compensated energy to achieve stationary music and a dynamic sweet spot. This system was adapted for a simple first person shooter VR game, and pilot tests were conducted to test its impact on the user experience.

1 Introduction

In audio post-production of VR games or 360-video experiences, spatial audio content including background music and sound effects are typically leveraged to keep users engaged in the experience. Sounds that are not in view such as a car driving behind the user may be used to trigger head movement. VR developers can thus take advantage of sound localization to direct the user's attention, so as to deliver artistic intentions effectively. Meanwhile, background music does not need to be controlled based on the head's movement in an artistic aspect unless it is the focus of the experience, as in soundscape recordings. Using head-tracking data and amplitude panning methods, the authors propose an audio reproduction system that provides user-interactive surround audio experience in a multi-channel loudspeaker configuration without headphones.

In a previous study, the authors proposed a method to anchor background music to the user's viewpoint in an evenly-spaced octagonal loudspeaker configuration that applies real-time panning [8].

Interactive panning over a multichannel loudspeaker layout can be achieved through several means. Vector-Base Amplitude Panning (VBAP) allows a sound source to be positioned anywhere in the space between two adjacent loudspeakers by adjusting speaker gains under the condition that the sum of the squares of the gain factors is constant [1]. When using VBAP, a virtual sound source vector and a loudspeaker vector, oriented from the listener's position, are calculated to obtain a gain vector containing gain coefficients for two loudspeakers located on either side of the virtual sound source. An alternative panning algorithm is Distance-Based Amplitude Panning (DBAP), which uses all the available speakers instead of a pair at a time [2]. DBAP calculates gain factors for all loudspeakers depending on the distance between each loudspeaker and the listener. It is similar to VBAP in that gain factors are derived from the speaker positions, but is also different since it requires the distances between loudspeakers and a virtual sound source, rather than directional vectors originating from the listener position [3][7]. Each gain factor is hence independent of the listener's position, as shown in Figure 1.

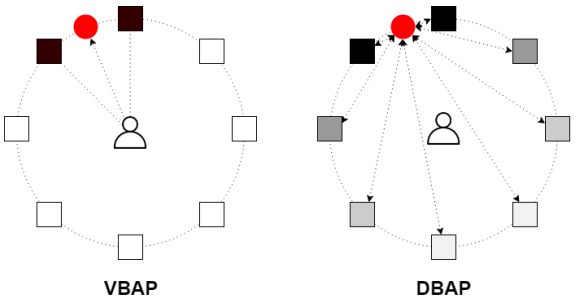


Figure 1. A virtual sound source (red circle) at -22.5° azimuth angle in octophonic layout. The color gradient of each loudspeaker (square) indicates the required gain coefficient for implementing VBAP or DBAP.

After the amplitude panning process, the sound pressure level and amount of delay for each loudspeaker can be modified appropriately to mimic the sweet spot, by tracking the user position. To address the main objective of making background music stationary in users' viewpoint in real-time, eight virtual sound sources can be represented by DBAP in an octophonic configuration in order to simulate 8.1 loudspeaker channels interacting with the user's viewpoint.

In the next section, other components of the proposed system such as SPL compensation, delay modification, and UnityOSC are described.

2 Implementation

An evenly-spaced octagonal loudspeaker layout (Front Center, Left, Right, Left Surround, Right Surround, Left Side, Right Side, Center Rear) in the horizontal layer is chosen for this study because of its relatively small separation angle between two loudspeakers (45º), which ensures that the phantom sound sources are more stable. Also, since the user is not stationary, an evenly spaced layout is desirable to ensure a consistent experience from all perspectives.

In this system, the music is converted from a stereo to a custom 8.1 output setup in real-time using a modified version of the up-mixing patch presented in [6]. A stereo mixed background music is used as an input signal, and up-mixed to 8 channels. This is achieved by modifying the typical 5.1 surround sound configuration output and adding back, side left, and side right channels to obtain an octagonal configuration.

Figure 2. Screenshots of main menu and playing mode of Shooter 360

An open source VR game (Figure 2), called "Shooter 360", is used to test the proposed audio reproduction system, in which a user aims and shoots at targets spawned at random locations in 360° space [10]. Several functionalities were added to the Max patch to provide a suitable audio experience over the octagonal loudspeaker configuration. Head-tracking information and sound FX-trigger data associated with the gameplay are communicated from Unity to Max/MSP. This information is then used to trigger sound effects at appropriate spatial positions, and to pan the background music.

2.1 Workflow

The prototype system consists of 3 main components- The Oculus Rift, UnityOSC UDP connection, and the audio signal processing in Max/MSP as shown in Figure 3. During the experience, the user wears the headset and is surrounded by the octagonal loudspeaker configuration.

The Oculus Rift head tracking and positional data is collected using the Unity Oculus SDK in real-time, which represents the user's viewpoint in terms of azimuth angle and physical position. Using the UnityOSC framework [5], this data is packaged as Open Sound Control (OSC) messages and sent via User Datagram Protocol (UDP) to the Max patch at fixed intervals of time. In addition to the user tracking data, gameplay information is also transmitted using the same framework, which is discussed in greater

detail in the following sections. Since the data is communicated via UDP, the Max patch can be used on the computer running the VR game or a different one.

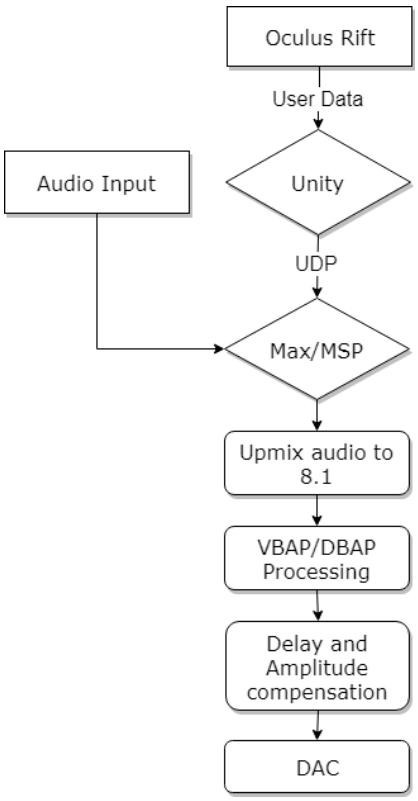


Figure 3. A flowchart of the real-time stationary music playback system for VR

The head tracking data received in Max determines the angle by which the background music needs to be rotated to remain anchored, and is used to calculate the gain values of the speakers using VBAP or DBAP. This yields the required azimuth shift for the virtual sources. Using one of the above panning algorithms, the gain factors of the loudspeakers can be calculated to anchor the music successfully. Once the speaker gain values are calculated, the 8 channels of music data are routed to the DAC, providing the anchored music experience.

For the anchored background music, the reference direction of the user's gaze is the center channel (0º azimuth). Here each channel is a virtual source that can be positioned according to the user's gaze. However, the angular distance between adjacent sources is always 45º; in other words, the eight virtual sources are always panned in unison, as shown in Figure 4.

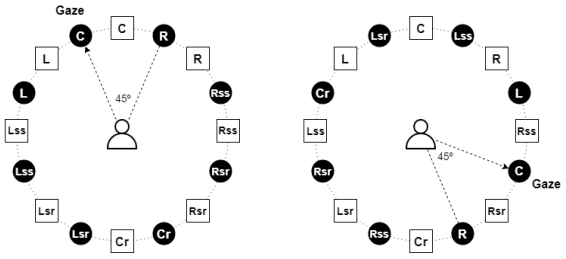


Figure 4. Phantom images move depending on listener's gaze direction in an octagonal loudspeaker configuration. The squares (white) and circles (black) represent physical loudspeakers and virtual sound sources, respectively. Arrows indicate the gaze direction, which point to a virtual sound source reproducing the center channel.

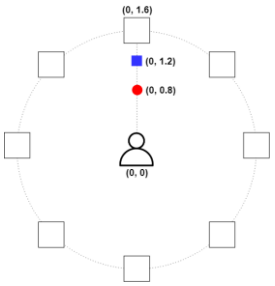
In addition, users can move around within a loudspeaker configuration. Thus, distances between the user and each loudspeaker are calculated in real-time so as to add varying amounts of delay and energy compensation to mimic the sweet spot located at the center of the loudspeaker configuration.

2.2 Amplitude Compensation

The amplitude of the audio channels can be compensated depending on the listener's position so as to mimic the sweet spot (center of the loudspeaker array). First of all, all loudspeaker positions are initially specified XY coordinates. In the experiment, the eight loudspeakers are evenly positioned in a circle with a radius of ~1.6 m. Assuming listener's position is (0, 0) in XY coordinates, center loudspeaker position is (0, 1.6). During the game, a player might move around within the loudspeaker configuration. To maintain a dynamic sweet spot (0, 0), sound pressure levels of each channel are compensated by using the inverse-square law:

$$L_d = 20\log(\frac{d_2}{d_1}), \quad (1)$$

where d_2 and d_1 are initial distance from user to a loudspeaker and new distance, respectively, and L_d is loudness difference. For example, if the user moves half a radius forward to the center loudspeaker, calculated loudness difference 6dB is subtracted from the current amplitude of the center loudspeaker, and corresponding compensated amplitudes are calculated for all the loudspeakers. This provides the user an illusion of being in the sweet spot irrespective of their movement. Figure 5 shows the compensated amplitude for each loudspeaker when the listener is at (0, 0.8) and (0, 1.2).



	(0, 0.8)	(0, 1.2)
L	-2.65	-2.99
R	-2.65	-2.99
C	-6.02	-12.0
Lss	+0.969	+1.938
Rss	+0.969	+1.938
Lsr	+2.916	+4.188
Rsr	+2.916	+4.188
Cr	+3.522	+4.861

Figure 5. Compensated amplitude (dB) for each channel at listener's positions (0, 0.8) and (0, 1.2).

Free space is an essential precondition for the inverse square law since the reflections coming from the enclosing surfaces, such as a wall and a ceiling, affect the way sound level decreases with distance. Reflected sound energy would change the "6 dB per distance double" rule closer to 4 dB rather than 6 dB. However, limited free space can even exist in the region close to the loudspeaker [11] if the direct sound is dominant and reflected sound is negligible, depending on the acoustics of the listening environment.

2.3 Delay Modification

Similar to the amplitude compensation method mentioned above, delay for each channel can be added to create a dynamic sweet spot by calculating the distance differences from the listener to each loudspeaker. Sample delay for the i^{th} loudspeaker is calculated as:

$$\frac{D_{max} - D_i}{343\ m/s} \cdot sr, \quad (2)$$

where D_i is distance between the listener and i^{th} loudspeaker, D_{max} is the maximum user-loudspeaker distance, 343 m/s is the speed of sound at 20°C, and sr is the sampling rate. Table 1 shows the modified delay time and samples for each loudspeaker when the listener is at (0, 0.8) and (0, 1.2) in the tested listening environment, as an example.

	● (0, 0.8)		■ (0, 1.2)	
	Time (ms)	Samples	Time (ms)	Samples
L	3.560	170	4.859	233
R	3.560	170	4.859	233
C	4.665	223	6.997	335
Lss	1.782	85	2.332	111
Rss	1.782	85	2.332	111
Lsr	0.471	22	0.608	29
Rsr	0.471	22	0.608	29
Cr	0	0	0	0

Table 1. Delay time and samples for each channel when listener's position is at (0, 0.8) and (0, 1.2).

2.4 Trigger Implementation for Sound FX

Using a UDP connection, head tracking data and trigger data for sound effects are transmitted from Unity to Max as shown in Table 2. First three data values provide the user's position (euler angle & XY coordinates). The other values serve a boolean-like function that only receives 0 (stop) and 1 (play). While data values 1-3 in the table are generic, 4-10 are specifically chosen to implement this VR game, and would need to be modified for other experiences. Based on these trigger values received during gameplay, sound effects are triggered to deliver an immersive experience. Some sound effects are always reproduced at virtual center, i.e. in the user's gaze direction (stationary), while the target spawn sounds are localized depending on their position in the game (interactive).

#	Description	Data Type	Audio Reproduction
1	User's gaze direction	Float	
2	User's X position coordinate	Float	
3	User's Y position coordinate	Float	
4	Trigger for weapon	Int	Stationary
5	Trigger for target hit	Int	Stationary
6	Trigger for menu gaze	Int	Stationary
7	Trigger for menu selection	Int	Stationary
8	Trigger for target spawn	Int	Interactive
9	New target's azimuth angle based on user's position	Float	
10	Trigger for background music	Int	

Table 2. OSC message contents

2.5 Graphical User Interface

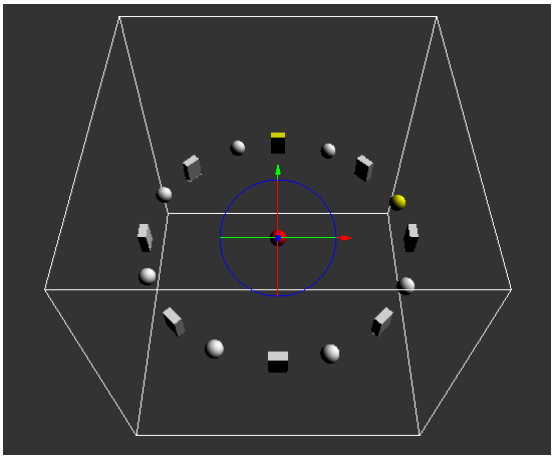


Figure 6. In the GUI, virtual sound sources interact with the user's gaze direction (67.5º). The yellow hexahedron and sphere indicate the physical loudspeaker and the virtual sound source for the center channel, respectively.

In the prototype, a graphical user interface was also created to track the movement of the virtual sound sources during the game as shown in Figure 6. The head-rotation data provides the required azimuth angle shifts for each phantom image, to enable eight virtual sound sources to move along with the user's gaze direction in real-time.

3 Discussion & Future Work

A multichannel audio reproduction system was created and successfully prototyped for a simple VR game. Pilot tests indicate that the system provides a seamless and immersive audio experiences to the users. There are, however, some additional features that could be added to improve the experience further.

A vertical sound image is required for audio reproduction system for VR to localize sound sources in three-dimensional space. In this system, only the horizontal plane was considered for audio reproduction. To address the main objective of simulating virtual sound sources in a 3D environment, additional loudspeakers could be added in the height layer.

A subjective study will be conducted to evaluate the perceptual impact of this system on the user experience, using a multichannel loudspeaker configuration without head-tracking as the reference. The effect of the stationary music system on localization accuracy of sound sources will be evaluated, in addition to the user preference and perceptual attributes such as immersion, timbral quality, sound image width, and clarity.

As a next step, a method using fewer loudspeakers for implementing stationary music and also for sound effects in 360º, i.e. including the vertical plane, will be explored. To do so, future work might involve the addition of height channel information using Vertical Hemispherical Amplitude Panning (VHAP) in which a phantom image can be elevated over the virtual upper hemisphere by configuring interchannel level differences among four horizontally positioned loudspeakers [9].

References

[1] Pulkki, V. "Virtual sound source positioning using vector base amplitude panning," *Journal of the Audio Engineering Society, 45(6),* pp. 456-466, 1997.

[2] Roginska, A., & Geluso, P. (Eds.) *Immersive Sound: The Art and Science of Binaural and Multi-channel Audio*. Taylor & Francis, pp. 251-252, 2017.

[3] Kostadinov, D., Reiss, J. D., & Mladenov, V. "Evaluation of distance based amplitude panning for spatial audio," *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 285-288, 2010.

[4] Pulkki, V. "Generic panning tools for MAX/MSP," *International Computer Music Conference,* 2000.

[5] Martin, J. G. UnityOSC: Open sound control classes and API interface for Unity 3D (Version 1.2) [Software]. Available from https://github.com/jorgegarcia/UnityOSC, 2012.

[6] Kim, S. "Subjective evaluation of stereo-9.1 upmixing algorithms using perceptual band allocation," *Audio Engineering Society Convention 145,* pp. 1-10, 2018.

[7] Lossius, T., Baltazar, P., & de la Hogue, T. "DBAP–distance-based amplitude panning," *International Computer Music Conference*, 2009.

[8] Kim, S., & Sridhar, S. "Stationary music from users' viewpoint in VR applications," *Audio Engineering Society Convention 145, 2018.*

[9] Lee, H., Johnson, D., & Mironovs, M. "Virtual hemispherical amplitude panning (VHAP): A method for 3D panning without elevated loudspeakers," *Audio Engineering Society Convention 144*, 2018.

[10] https://assetstore.unity.com/packages/essentials/tutorial-projects/vr-samples-51519

[11] Everest, F. A. *Master Handbook of Acoustics*. McGraw-Hill, pp.83-88, 2001.