Audio Engineering Society

Convention e-Brief

Presented at the 137th Convention
2014 October 9–12 Los Angeles, CA, USA

Dynamic Audio Imaging in Radial Virtual Reality Environments

Mark Blewett, James Pinkl, and Bryan Dalle Molle

Electronic Visualization Laboratory, University of Illinois at Chicago, Chicago, IL, 60607, USA
markblewett@hotmail.com
jimmypinkl@yahoo.com
bryandallemolle@gmail.com

ABSTRACT

The CAVE2 is a large-scale, 320-degree, 3D/2D, virtual-reality environment developed by the Electronic Visualization Laboratory at the University of Illinois at Chicago (Figure 1). The environment has a tracking system that determines the location of the primary user, and then renders 3D images to his perspective. The CAVE2 also has a 20.2 channel sound system controlled by a SuperCollider-based audio server that is, in turn, controlled via the Omicron SoundAPI. Audio imaging for the system had operated with the assumption that the listener was stationary in the center of the CAVE2 and the individual source width of audio objects was not dynamically altered. Our team added functionality to maintain audio imaging, including source position and width, as a tracked user and sound object move relative to one another within the environment.

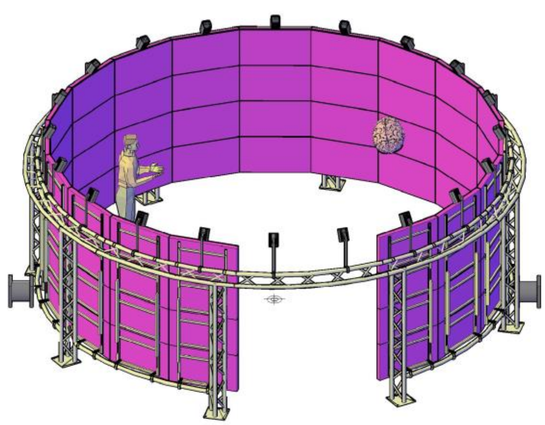

Figure 1 CAVE2 as located at the University of Chicago at Illinois' Electronic Visualization Laboratory

1. MOTIVATION

CAVE2 had in place a large multichannel sound system consisting of 20 two-way, near-field reference monitors equally spaced around the 6.8 meter circumference of the environment, as well as a pair of companion subwoofers. The system was fed from a SuperCollider audio server that provided sound objects with inverse distance law attenuation while utilizing the PanAz UGen [1] to position them in the plane, both parameters being derived relative to the center of the CAVE2. The server, in turn, received instruction from OmegaLib, the CAVE2's custom middleware, through the aforementioned Omicron SoundAPI. From the designer's perspective, scenes were most easily authored for OmegaLib's Python API. It was determined that this unique environment and its existing infrastructure provided opportunities to improve the spatial accuracy of its sound system to better accompany the excellent stereophonic visual experience.

The first step was to use the existing tracking technology to maintain audio imaging, position and width, relative to the tracked user moving about the environment. Additionally, to reduce distortion of the image as the user moves within the sound field, the PanAz UGen was to be modified to calculate an amplitude weighting factor based on the user's proximity to each loudspeaker. It is important to note here that this step was not intended to optimize the implementation of any of the various techniques for creating perceived object localization (e.g. HOA, WFS, VBAP, etc.), or decorrelating width channels [2][3]. Rather, the existing tools within SuperCollider, in this case the PanAz function, were used until further experiments are completed.

2. MAINTAINING AUDIO IMAGING

2.1. Source Position

In a 3D virtual world, three sets of coordinates are necessary to determine the relative position of objects to a mobile user within a static speaker array. First, an object must be placed within the virtual world (world position). Next, the center of CAVE2 in the world must be specified and all objects' locations relative to this point ascertained (local position). Finally, we must know the position of the user within the CAVE2 (user position).

With knowledge of an audio object's local position (*O*), the user position (*U*), and the radius of the environment (*r*), the position of an audio source relative to the mobile user (source position = *S*) can be specified. We first determine the equation of the line using the object and the user position coordinates. This line intersects the circumference of the CAVE2 at two points (I_1, I_2). We solve this simple quadratic equation and obtain its two solutions (Figure 2). However, when we attempt to determine which solution is the correct one, we encounter three cases.

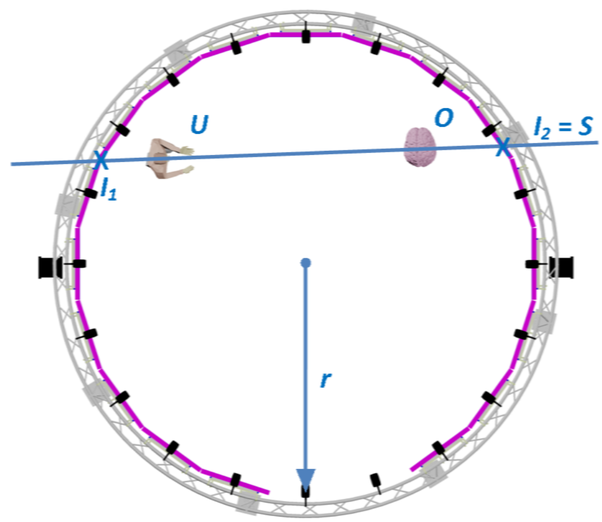

Figure 2 Line ($\overline{UO}$) between user and sound object, showing wall intercepts (I_1, I_2) and correct source position (S_1).

In the first case, the magnitude of the local position vector of the audio object equals the radius of the CAVE2; i.e., the object is located on the walls of the environment. Here the solution is trivial and the source position coincides with the local position of the object.

In the second case, the magnitude of the local position vector of the object is less than the CAVE2 radius; i.e., the object is located within the walls of the environment. Here, the correct solution to the quadratic will be on the far side of the object. Equivalently, the magnitude of the two dimensional vector from the source position to the object will be smaller than the magnitude of the two-dimensional vector from the source position to the user.

The third case arises when the object is outside of the environment's walls. In this case we choose the solution that is closest to the object.

Following this logic, the Omicron SoundAPI was modified to compute the source position dynamically. The current source position, in radians over pi, is sent via an OSC message to the audio server. This message then serves as the "pos" argument to the modified PanAz UGen, which pans the object around the circumference of the CAVE2.

2.2. Source Width

The width of an audio object is inversely proportional to the observer's distance from it [4]. Again utilizing the tracked user position and local position of the object, we can dynamically alter a user-defined object width. To begin, the designer is free to define an audio object's width on a scale of 1 to 20, where 1 corresponds to a phantom image panned between two adjacent speakers and 20 corresponds to a sound being played from all 20 loudspeakers. Here, the object's width is, arbitrarily, defined at one meter's distance. The spatial extent of the object is then dynamically altered by multiplication by the reciprocal of the distance between the object and the user. The function is stabilized by limiting the dynamic width to values between 1 and 20 and by maintaining a minimum object to user distance of 0.25 meters (1); these values having been informally selected as providing a reasonably natural response.

$$1 \le W_{dynamic} = \frac{W_{defined}}{\left|\overrightarrow{UO}\right|} \le 20, \left|\overrightarrow{UO}\right| \ge 0.25m \quad (1)$$

Again, this feature was implemented in the Omicron SoundAPI. The current width is sent to the audio server via an OSC message where it serves as the "width" argument to the modified PanAz UGen.

2.3. Loudspeaker Amplitude Weighting

Now that a procedure is established to maintain the basic audio image of objects as they and a user move relative to one another in the environment, we must compensate for the varying distances between the mobile user and each of the individual, static loudspeakers. Currently under investigation, our method to normalize the loudspeaker amplitudes is to find the square of the ratio of the distance between the user and each particular loudspeaker over the radius of CAVE2 (Figure 3). Then, this ratio is used as a weighting factor for the corresponding audio channel's amplitude (2). This inverse square law approximation will be used as an initial starting point until further experiments can be completed (see section 3).

This time, the logic is implemented on the server side in SuperCollider via a quark extension. Using the popular PanAz UGen as a basis, this plugin takes additional arguments: the circular coordinates of the user (the angle in radians/pi) and the radius of the environment. The azimuthal panning is utilized "as-is", but each output channel is now assigned the weight discussed above. In this way, the integrity of the audio image is independent of the user's location in the CAVE2.

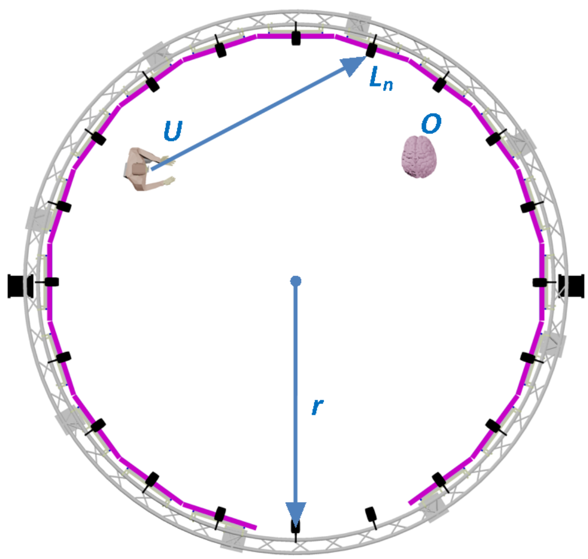

Figure 3 The distance between the user and loudspeaker is divided by the radius of the CAVE2. This ratio is squared to find that loudspeaker's amplitude weighting.

$$L_{newAMP} = L_{oldAMP} \left(\frac{\left|\overrightarrow{UL_n}\right|}{r} \right)^2 \quad (2)$$

3. FURTHER WORK

Basic audio imaging is now maintained in CAVE2 as the user moves about the environment, but there is still much to do to refine these features. Some informally derived values, such as those used in the source width feature, must be formally evaluated with quantitative measurements and/or qualitative user studies. The various methods for establishing source location must be evaluated. And, source width channels must be decorrelated in a manner that does not adversely affect the timbral qualities of the objects [2][3].

Additionally all current processes assume free field conditions, even though the CAVE2 is a highly reflective, enclosed environment. Therefore, we plan to map the transfer function of the environment. First we will use it to determine a more realistic curve for the loudspeaker amplitude weighting function. Then we hope to use it to improve the realism of audio experience by constructing a dynamic inverse filter capable of minimizing the effect of the space on the listening experience. This will aid designers by maximizing the portability of their creations while simultaneously reducing the imposed sonic signature of CAVE2.

Other projects seek to expand the feature set of the audio server in an effort to increase the level of realism possible in CAVE2 soundscapes. We aim to model, for example, room reflections, air absorption, Doppler effects, and even occlusion of one virtual object by another.

4. ACKNOWLEDGMENTS

This work was supported by the Electronic Visualization Laboratory at the University of Illinois at Chicago. Graphics were drafted by Jonas Talandis. Special thanks to Maxine Brown, Andrew Johnson, Lance Long, Arthur Nishimoto (Omicron), and Alessandro Febretti (OmegaLib). This work builds on the original CAVE2 audio server developed by JD Pirtle. Also, thanks to Vahe Caliskan, UIC Department of Electrical and Computer Engineering, and Zachary Seldess and Michelle Daniels of Sonic Arts Research and Development, UCSD, for additional support. CAVE2 is a trademark of the University of Illinois Board of Trustees.

5. REFERENCES

[1] Reference to SuperCollider help file URL: https://github.com/supercollider/supercollider/blob/master/HelpSource/Classes/PanAz.schelp

[2] Potard, G & Burnett, I (2003): “A Study on Sound Source Apparent Shape and Wideness”. In E. Brazil & B. Shinn-Cunningham (Eds.), Proceedings of the 2003 International Conference on Auditory Display (pp. 25-28). Boston, USA: Boston University Publication Production Department.

[3] T. Pihlajamaki, O. Santala, and V Pulkki (2014), “Synthesis of Spatially Extended Virtual Sources with Time-Frequency Decomposition of Mono Signals”. Journal of the Audio Engineering Society, Vol. 62, No. 7/8, pp. 467-484

[4] Hyunkook, Lee (2012): “Subjective Evaluatioins of Perspective Control Microphone Array (PCMA)”. Audio Engineering Society 132nd Convention, Convention Paper 8625