A Web-based System for Designing Interactive Virtual Soundscapes

Anıl Çamcı, Paul Murray and Angus G. Forbes
University of Illinois at Chicago
Electronic Visualization Lab
[acamci, pmurra5, aforbes]@uic.edu

ABSTRACT

With the advent of new hardware and software technologies, virtual reality has gained a significant momentum recently. VR design tools, such as game engines, have become much more accessible and are being used in a variety of applications ranging from physical rehabilitation to immersive art. These tools, however, offer a limited set of tools for audio processing in 3D virtual environments. Furthermore, they are platform-dependent due to performance requirements and feature separate editing and rendering modes, which can be limiting for sonic VR implementations. To address these, we introduce a novel web-based system that makes it possible to compose and control the binaural rendering of a dynamic open-space auditory scene. Developed within a framework of well-established theories on sound, our system enables a highly detailed bottom-up construction of interactive virtual soundscapes by offering tools to populate navigable sound fields at various scales (i.e. from sound cones to 3D sound objects to sound zones). Based on modern web technologies, such as WebGL and Web Audio, our system operates on both desktop computers and mobile devices. This enables our system to be used for a variety of mixed reality applications, including those where users can simultaneously manipulate and experience a virtual soundscape.

1. INTRODUCTION

Sound is an inherently immersive phenomenon. The air pressure originating from a sound source propagates in three dimensions. Although music is considered primarily a temporal art, the immersive quality of sound has been exploited throughout music history: in ancient antiphons, different parts of the music were sung by singers located at opposing parts of a church to amplify the effect of the call-and-response structure[1]. In the 1950s, the composer Karlheinz Stockhausen composed one of the first pieces of quadraphonic music using a speaker placed on a rotating table surrounded with 4 microphones. When played back, the resulting recording would envelope the listener with swirling gestures. Since the 1950s, many sound art pieces have highlighted the spatial qualities of sound by exploring the continuities between music and other art forms such as painting and sculpture.

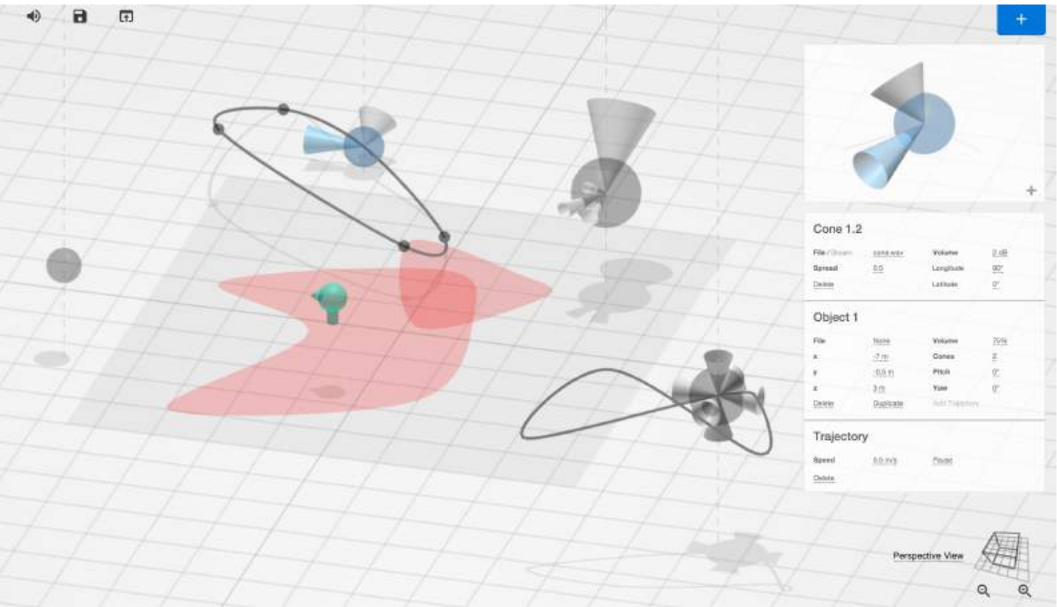
In recent years, immersive media has been gaining popularity with the advent of new technologies such as commercial depth-tracking devices and head-mounted displays. Accordingly, software tools to create immersive media has become more accessible. Many artists today, for instance, use game engines to create virtual reality artworks. However, modern immersive design tools heavily favor the visual domain. Despite many studies that have highlighted the role of audio in improving the sense of immersion in virtual realities [2, 3], audio processing in modern game engines remain an afterthought. We have previously discussed a sound-first VR approach based on well-established theories on sound objects and soundscapes [4]. Building up on the taxonomy introduced in this study, the current paper introduces a novel web-based system that enables the rapid design of both virtual sonic environments and the assets (i.e., sound objects and sound zones) contained within them. Specifically, our system:

- provides a user-friendly 3D environment specific to sonic virtual realities, with specialized components such as sound objects and sound zones;
- offers both interactive and parametric manipulation of such components, enabling a precise control over highly-detailed virtual soundscapes;
- introduces a multi-cone model for creating 3D sound objects with complex propagation characteristics;
- enables adding dynamism to objects via hand-drawn motion trajectories that can be edited in 3D;
- makes it possible to manipulate virtual sonic environments at various scales using multiple view and attribute windows;
- offers a unified interface for the design and the simulation of such environments, allowing the user to modify a sound field in real-time;
- operates on the web-browser so that it supports mobile devices, which therefore makes it possible for the user to simultaneously explore and edit augmented sonic realities.

2. RELATED WORK

2.1 Sound in Virtual Reality

Modern VR design tools, such as game engines, offer basic audio assets, including point sources and reverberant zones. These objects are created and manipulated through



Cone 1.2

File / Stream cone.wav

Volume 2 dB

Spread 0.5

Longitude 90°

Latitude 0°

Object 1

File None

Volume 75%

x -7 m

Cones 2

y -0.5 m

Pitch 0°

z 3 m

Yaw 0°

Trajectory

Speed 0.5 m/s

Figure 1. A screenshot of our user interface on a desktop computer displaying an object with two cones and a motion trajectory being edited. On the top right region, a close-up window displays the object with the cone that is currently being interacted with highlighted in blue. The windows below this close-up allows the user to control various attributes of the cone, the parent object, and its trajectory. Two overlapping sound zones are visualized with red polygons. A gray square represents the room overlay. The user is represented with a green dummy head.

the same interactions used for visual objects on these platforms. Additionally, third-party developers offer plug-ins, such as 3Dception[1], Phonon 3D[2] and RealSpace3D[3], that extend the audio capabilities of these engines with such features as occlusion, binaural audio, and Ambisonics. However, these extensions act within the UI framework of the parent engine, and force the designer to use object types originally meant to describe graphical objects, which can be limiting for sound artists.

Other companies specialize in combined hardware and software VR solutions. *WorldViz*, for instance, offers an “Ambisonic Auralizer” consisting of a 24-channel sound system, which can be controlled with Python scripts using their VR design platform called Vizard[4]. Although their tools have powerful spatializing capabilities, no user interfaces exist for creating sonic environments using them.

The *Zirkonium* software developed initially for the *Klangdom* surround sound system at the ZKM Institute for Music and Acoustics, allows the design of multiple spatial trajectories for sound sources [5]. Furthermore, the software allows the parametric and temporal manipulation of these trajectories.

IRCAM’s *Spat* software[5] enables the creation of dynamic 3D scenes using binaural audio and Ambisonics. Although *Spat* provides a comprehensive set of tools which can be used to develop 3D audio applications within the Max programming environment, it does not offer a singular interface for virtual environment design.

SoundScape Renderer [6], developed by researches at the Quality and Usability lab at TU Berlin, is a system for the positioning of sound sources around a stationary listener using a 2D overview of the scene. Users of this software can assign arbitrary sound files and input sources to virtual objects. The SoundScape Renderer offers advanced rendering techniques, such as WFS, VBAP, Ambisonics as well as binaural audio.

2.1.1 Web Audio API

The Web Audio API [7] is a JavaScript library for processing audio in web applications. A growing number of projects utilize this tool due to its high-level interface and its ability to operate on multiple platforms. Using the Web Audio API, Rossignol et al. [8] designed an acoustic scene simulator based on the sequencing and mixing of environmental sounds on a time-line. Lastly, Pike et al. [9] developed an immersive 3D audio web application using head-tracking and binaural audio. The system allows its users to spatialize the parts of a musical piece as point sources in 3D. These examples demonstrate that Web Audio is powerful enough to be used as a back end for sonic virtual realities.

Our implementation utilizes the built-in binaural functionality of the Web Audio API, which is derived from *IRCAM Listen*’s head-related transfer function (HRTF) database [10]. However, several studies have shown that non-individualized HRTFs yield inconsistent results across listeners in terms of localization accuracy [11]. Although the Web Audio API does not currently support the use of custom HRTFs, recent studies have shown that it can be extended to allow users to upload individualized HRTFs [10, 9].

2.1.2 Virtual Acoustic Environments

Studies on virtual acoustic environments (VAEs) investigate the modeling of sound propagation in virtual environments through source, transmission, and listener modeling [12]. In the 1990s, Huopanemi et al. [13] developed DIVA Virtual Audio Reality System as a real-time virtual

[1] https://twobigears.com/3dception.php

[2] https://www.impulsonic.com/products/phonon3d/

[3] http://realspace3daudio.com

[4] http://www.worldviz.com/products/vizard

[5] http://forumnet.ircam.fr/product/spat-en

audiovisual performance tool with both hardware and software components. The system used MIDI messages to move virtual instruments in space using binaural rendering.

A commercial application of VAEs is the simulation of room acoustics for acoustic treatment purposes. In such applications, a specialized software allows the users to load architectural models and surface properties to simulate propagation characteristics of sound within a given space, such as a concert hall, theatre, office, or a restaurant. In a basic auralization (or sound rendering) pipeline used in VAEs, the acoustic model of a virtual environment is used to filter an audio signal to create an auditory display in the form of a spatialized signal [14, 15]. While previous projects have offered efficient methods for the rendering of virtual acoustic environments [16, 17, 18], it remains a challenging task to compute a high-density sonic environment with acoustic modelling, as the computational load depends linearly on the number of virtual sources [17].

3. OVERVIEW OF THE SYSTEM

A system for the computational design of virtual soundscapes requires audio-to-visual representations. In digital audio workstations, a sound element is represented by a horizontal strip that extends over a timeline, where the user can edit a single sound element by cutting and pasting portions of this strip. Furthermore, multiple strips can be aligned vertically to create simultaneous sound elements. However, in the context of a virtual reality application, conceiving sound elements as spatial entities, as opposed to temporal artifacts, requires a different framework. To represent the different components of spatialized sound, we use visual elements— such as spheres, cones, splines and polygons— that are more applicable to the spatial composition of a sonic environment.

Based on the JavaScript library Three.js, our system utilizes a 3D visual scene, which the user can view at different angles to edit the layout of objects. However, manipulating and navigating an object-rich 3D scene using a 2D display can get complicated. Previous work has shown that, in such cases, using separate views with limited degrees of freedom is faster than single-view controls with axis handles [19]. Accordingly, in our system, the 2D bird's-eye view allows the user to manipulate the position of components on the lateral plane, while the 3D perspective view is exclusively used to control the height of the objects or trajectory control points.

We provide a unified environment for designing both open-space sonic environments and the sound objects contained within them. We combined a multiple-scale design [20] with a dual-mode user interface [21], which improves the precision at which the user can control the various elements of the virtual soundscape, from sound cones to sound objects to sound fields. We also utilized dynamic attribute windows to offer parametric control over properties that are normally controlled via mouse or touch interactions. This enables a two-way interaction between abstract properties and the virtual environment in a combined design space [22], which is used in information-rich virtual environments such as ours.

Furthermore, our system allows the user to simultaneously design and explore a virtual sound field. In modern game engines, the editing and the simulation phases are often separated due to performance constraints. However, since our underlying system is designed to maintain an audio environment, which is computationally less demanding than graphics-based applications, editing and navigation can be performed concurrently.

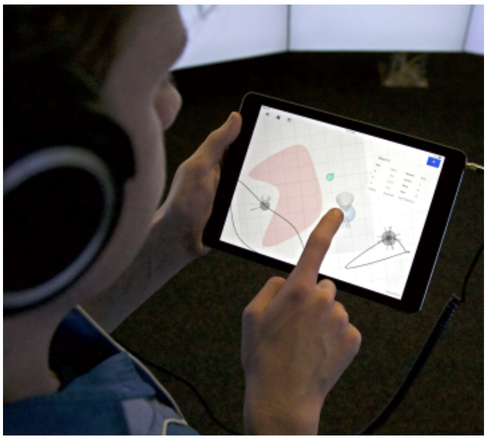


Figure 2. A user exploring the augmented reality in a CAVE system, while using a mobile device to edit the 3D sonic virtual reality he is hearing through headphones. The user is controlling the position of an object in lateral-view mode.

Finally, we offer an amalgamation of virtual and augmented reality experiences for the user. Given the ability of our system to function both on desktop and tablet computers, the user of an augmented reality implementation can manipulate the virtual environment using a mobile device while exploring the physical space onto which a virtual soundscape is superimposed, as seen in Fig. 2.

4. SOUND FIELD

The *sound field* is the sonic canvas onto which the user can place a variety of components, such as sound objects and sound zones. In the default state, the sound field is represented by a 2D overhead-view of an infinite plane. The user can zoom in and out of the sound field and pan the visible area. Furthermore, the sound field can be tilted and rotated. Whenever the user interacts with the sound field to add a new sound object, zone or trajectory, the view automatically switches to the bird's-eye view to allow for object placement. The user can then switch to the perspective view by clicking the view indicator on the bottom right corner of the interface. A global mute button allows the user to turn off the entire audio output. This makes it possible to make offline editions to the sound field. Furthermore, with dedicated icons found adjacent to the mute button, the user can save and load system states to restore a previously composed sound field.

4.1 Navigating the Interactive Virtual Soundscape

The user can explore the virtual sonic environment via one of two modalities, or a combination of both. In *virtual navigation*, a stationary user is equipped with a headphone connected to the device running the system. Using keyboard controls, the user can travel within the sound field virtually. In *augmented navigation*, the user moves physically within a room that is equipped with a motion-tracking system. User's gaze direction is broadcasted to the system via OSC to update the position and the orientation of the Web Audio's *Listener Node*, which effectively controls the binaural rendering of the auditory scene based on the user's movements.

In augmented reality applications of our system, the user can define a sub-plane within the sound field to demarcate the region visible to the motion-tracking system. The demarcated region is represented by a gray translucent polygon on the sound field. The users can adapt the room overlay to the particular room they are in by mapping the vertices of this polygon to the virtual positions tracked when they are standing at the corners of the room. Sound components can be placed inside or outside the boundaries of the room.

5. SOUND OBJECTS

5.1 Multi-cone implementation

In modern game engines, users can populate a scene with a variety of visual objects. These objects range from built-in assets to 3D models designed with third-party software. Sound assets are phantom objects that define position and, when available, orientation for sound files that are to be played back in the scene. Sound assets can be affixed to visual objects to create the illusion of a sound originating from these objects. Directionality in game audio can be achieved using sound cones. A common implementation for this consists of two cones [7]: an inner cone plays back the original sound file, which becomes audible when the user's position falls within the projection field of the cone. An outer cone, which is often larger, defines an extended region in which the user hears a attenuated version of the same file. This avoids unnatural transitions in sound levels, and allows a directional sound object to fade in and out of the audible space.

However, sound producing events in nature are much more complex. Parts of a single resonating body can produce sounds with different directionality, spread, and throw characteristics. With a traditional sound cone implementation, the user can generate multiple cones and affix them to the same point to emulate this behavior, but from a UI perspective, this quickly gets cumbersome to design and maintain. In our system, we have implemented a multi-cone sound object that allows the user to easily attach an arbitrary number of right circular cones to a single object, and manipulate them.

5.2 Interaction

After pressing the *plus* icon on the top right corner of the UI, the user can click anywhere in the sound field to place a new sound object. The default object is an ear-level[6] omnidirectional point source represented by a translucent sphere on the sound field.

Creating a new object, or selecting an existing object, brings up an interactive close-up view, as seen in Fig. 3, as well as an attribute window on the top right region of the screen. The sound field view remains unchanged providing the user contextual control over the object that is being edited in the close-up window. The close-up view allows the user to add or remove sound cones and position them at different longitude and latitude values. Interacting with a cone brings up a secondary attribute window for local parameters, where the user can attach a sound file or an audio stream to a cone, as well as control the cone's base radius and lateral height values. The base radius controls the projective spread of a sound file within the sound field, while the height of a cone determines its volume. These attributes effectively determine the spatial reach of a particular sound cone. The secondary attribute window also provides parametric control over longitude and latitude values. Each object can be duplicated with all of its attributes. A global volume control allows the user to change the overall volume of an object, which is represented by the radius of the translucent sphere.

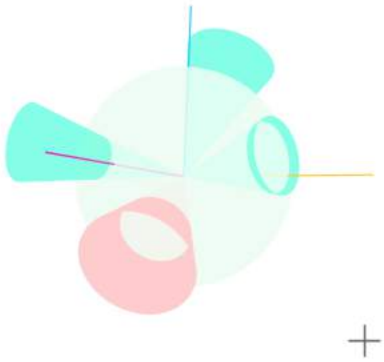


Figure 3. A screenshot of the object close up view displaying a sound object with four cones. The cone in red is currently being interacted with.

5.3 Trajectories

The user can attach arbitrarily drawn motion trajectories to each sound object. If the start and stop positions of a trajectory drawing are in close proximity, the system interpolates between these points to form a closed-loop trajectory. Once the action is completed the object will begin to loop this trajectory using either back-and-forth or circular motion depending on whether the trajectory is closed or not. Once a trajectory has been defined, a trajectory attribute window allow the user to pause, play, change motion speed in either direction or delete the trajectory. A resolution attribute allows the user the change the number of control points that define the polynomial segments of a trajectory curve. Once the user clicks on an object or its trajectory, these control points become visible and can be repositioned in 3D.

6. SOUND ZONES

For ambient sounds or sounds that are to be perceived as originating from the listener, we have implemented the sound zone component, which demarcates areas of non-directional and omnipresent sounds. Once the user walks into a sound zone, they will hear the source file attached to the zone without distance or localization cues.

[6] Ear-level is represented by the default position of the audio context listener object on the Y-axis.

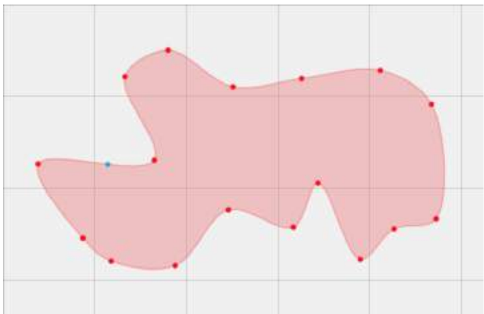


Figure 4. A screenshot of a sound zone that is being edited in the bird's-eye view mode. The user is about to add a new control point at the location highlighted with the blue dot.

6.1 Interaction

After clicking the *plus* icon on the top right corner, the user can draw a zone of arbitrary size and shape within the sound field with a click-and-drag action. Once the action is completed, the system generates a closed spline curve by interpolating between action start and stop positions. When a new zone is drawn, or when an existing zone is selected, a window appears on the top right region of the screen to display zone attributes, which include audio source, volume, scale, and rotation. An existing zone can be reshaped by adding new control points or moving the existing ones, as seen in Fig. 4.

7. APPLICATIONS

The ease of use, detail of control, and the unified editing and navigation modes provided by our system not only improve upon existing applications but also open up new creative and practical possibilities.

Interactive virtual soundscapes have many applications ranging from artistic practice to data sonification. As a compositional tool, our system constitutes a platform to create works consisting of sounds that act as spatial entities rather than events that are part of a temporal progression, which is often emphasized in modern digital audio workstations. Our system allows the composer to visualize sound sources located in space, and therefore have a better grasp of the spatial configuration of separate sound objects. Using our built-in objects, the composer can create complex sound morphologies, and layer a multitude of objects to explore spatially emergent sonic characteristics. Furthermore, the real-time design features of our system make it possible to use the system in concert situations, where the artist's construction of a virtual soundscape becomes a part of the performance. Furthermore, besides these uses that are intended for the artist, the listeners can also use our system to create casual spatial listening experiences.

Our system can also be utilized in sound pedagogy. Ear cleaning exercises, first proposed by R. Murray Schafer [23], aim at improving people's awareness of not only their immediate sonic environments, but also the precision with which they can listen to their surroundings. Ear cleaning exercises focusing on dynamic, spectral and spatial characteristics of environmental sounds can be administered using our system. Multi-participant exercises can be conducted using the augmented reality system. Furthermore, new ear cleaning exercises, such as re-spatializing of real-time audio input in the virtual soundscape, can be envisioned.

While our system relies on basic and widely-adopted mouse and touch interactions, it also affords a parametric control of object, zone and sound field attributes. This allows it to be utilized as a sonification tool for scientific applications, where researchers can rapidly construct detailed and accurate auditory scenes. Furthermore, since it can receive and transmit OSC data, our system can be interfaced with other software. This allows the control of sound objects via external controllers or data sets, but also enables the system to broadcast sound field data to other applications with OSC capabilities, such as Processing, openFrameworks and Unity.

Our system can also be used as an on-sight sketching tool by landscape architects to simulate the sonic characteristics of open-air environments. By mapping the target location on our sound field, the architect can easily construct a virtual environment with sound producing events within both the target location and the area surrounding it.

8. FUTURE WORK AND CONCLUSIONS

In the near future, we plan to implement 3D objects that enable sound occlusion. This implementation will allow the artist to draw non-sounding objects in arbitrary shapes that affect the propagation of sounds around them. Furthermore, although velocity functions that are used to achieve doppler effects has been deprecated in the recent version of the Web Audio API, we plan to add this feature to better simulate objects with motion trajectories. We also plan to improve the sound zone implementation with gradient volume characteristics. Similar to radial and linear gradient fill tools found in graphics editors, this feature will allow the user to create sound zones with gradually evolving amplitude characteristics. Additionally, we plan to implement features that will facilitate rich mixed reality applications. For instance, incorporating a video stream from the tablet camera will allow the user to superimpose a visual representation of the sound field onto a live video of the room they are exploring with a tablet.

In this paper, we introduced a novel system to design and control interactive virtual soundscapes. Our system provides an easy-to-use environment to construct highly-detailed scenes with components that are specialized for audio. It offers such features as simultaneous editing and navigation, web-based cross-platform operation on mobile and desktop devices, the ability to compose complex sound objects and sound zones with dynamic attributes that can be controlled parametrically using secondary attribute windows, and multiple views to simplify 3D navigation. As a result, our system provides new creative and practical possibilities for composing and experiencing sonic virtual environments.

9. REFERENCES

[1] R. Zvonar, "A History of Spatial Music: Historical Antecedents from Renaissance Antiphony to Strings in the Wings," *eContact*, vol. 7, no. 4, 2005.

[2] D. R. Begault, *3-D Sound for Virtual Reality and Multimedia*. San Diego, CA, USA: Academic Press Professional, Inc., 1994.

[3] F. Grani, S. Serafin, F. Argelaguet, V. Gouranton, M. Badawi, R. Gaugne, and A. Lécuyer, "Audio-visual Attractors for Capturing Attention to the Screens when Walking in CAVE Systems," in *IEEE VR Workshop on Sonic Interaction in Virtual Environments (SIVE)*, 2014, pp. 75–76.

[4] A. Çamcı, Z. Özcan, and D. Pehlevan, "Interactive Virtual Soundscapes: A Research Report," in *Proceedings of the 41st International Computer Music Conference*, 2015, pp. 163–169.

[5] C. Miyama, G. Dipper, and L. Brümmer, "Zirkonium Mk III: A Toolkit for Spatial Composition," *Journal of the Japanese Society for Sonic Arts*, vol. 7, no. 3, pp. 54–59.

[6] M. Geier and S. Spors, "Spatial Audio with the Sound-Scape Renderer," in *27th Tonmeistertagung–VDT International Convention*, 2012.

[7] P. Adenot and R. Toy. (2016) Web Audio API. [Online]. Available: http://webaudio.github.io/web-audio-api/.

[8] M. Rossignol, G. Lafay, M. Lagrange, and N. Misdarris, "SimScene: a Web-based Acoustic Scenes Simulator," in *Proceedings of the 1st Web Audio Conference*, January 2015.

[9] C. Pike, P. Taylour, and F. Melchior, "Delivering Object-Based 3D Audio Using The Web Audio API And The Audio Definition Model," in *Proceedings of the 1st Web Audio Conference*, January 2015.

[10] T. Carpentier, "Binaural Synthesis with the Web Audio API," in *Proceedings of the 1st Web Audio Conference*, January 2015.

[11] S. Zhao, R. Rogowski, R. Johnson, and D. L. Jones, "3D Binaural Audio Capture and Reproduction Using a Miniature Microphone Array," in *Proceedings of the 15th International Conference on Digital Audio Effects (DAFx)*, 2012, pp. 151–154.

[12] L. Savioja, J. Huopaniemi, T. Lokki, and R. Väänänen, "Creating Interactive Virtual Acoustic Environments," *J. Audio Eng. Soc*, vol. 47, no. 9, pp. 675–705, 1999.

[13] J. Huopaniemi, L. Savioja, and T. Takala, "DIVA Virtual Audio Reality System," in *Proceedings of International Conference on Auditory Display (ICAD)*, November 1996, pp. 111–116.

[14] T. Funkhouser, J. M. Jot, and N. Tsingos, ""Sounds good to me!"–Computational Sound for Graphics, Virtual Reality, and Interactive Systems," *ACM SIGGRAPH Course Notes*, pp. 1–43, 2002.

[15] T. Takala and J. Hahn, "Sound Rendering," *SIGGRAPH Computer Graphics*, vol. 26, no. 2, pp. 211–220, Jul. 1992.

[16] R. Mehra, A. Rungta, A. Golas, M. Lin, and D. Manocha, "WAVE: Interactive Wave-based Sound Propagation for Virtual Environments," *IEEE Transactions on Visualization and Computer Graphics*, vol. 21, no. 4, pp. 434–442, 2015.

[17] M.-V. Laitinen, T. Pihlajamäki, C. Erkut, and V. Pulkki, "Parametric Time-frequency Representation of Spatial Sound in Virtual Worlds," *ACM Transactions on Applied Perception (TAP)*, vol. 9, no. 2, p. 8, 2012.

[18] T. Yiyu, Y. Inoguchi, E. Sugawara, M. Otani, Y. Iwaya, Y. Sato, H. Matsuoka, and T. Tsuchiya, "A Real-time Sound Field Renderer Based on Digital Huygens' Model," *Journal of Sound and Vibration*, vol. 330, no. 17, pp. 4302 – 4312, 2011.

[19] J.-Y. Oh and W. Stuerzlinger, "Moving Objects with 2D Input Devices in CAD Systems and Desktop Virtual Environments," in *Proceedings of Graphics Interface 2005*, 2005, pp. 195–202.

[20] B. B. Bederson, J. D. Hollan, K. Perlin, J. Meyer, D. Bacon, and G. Furnas, "PAD++: A Zoomable Graphical Sketchpad for Exploring Alternate Interface Physics," *Journal of Visual Languages and Computing*, vol. 7, pp. 3–31, 1995.

[21] J. Jankowski and S. Decker, "A Dual-mode User Interface for Accessing 3D Content on the World Wide Web," in *Proceedings of the 21st International Conference on World Wide Web*, 2012, pp. 1047–1056.

[22] D. A. Bowman, C. North, J. Chen, N. F. Polys, P. S. Pyla, and U. Yilmaz, "Information-rich Virtual Environments: Theory, Tools, and Research Agenda," in *Proceedings of the ACM Symposium on Virtual Reality Software and Technology*. ACM, 2003, pp. 81–90.

[23] R. M. Schafer, *Ear Cleaning, Notes for an Experimental Music Course*. Toronto, CA: Clark & Cruickshank, 1969.