A Web-based UI for Designing 3D Sound Objects and Virtual Sonic Environments

Anıl Çamcı*, Paul Murray† and Angus Graeme Forbes‡
Electronic Visualization Laboratory, Department of Computer Science
University of Illinois at Chicago

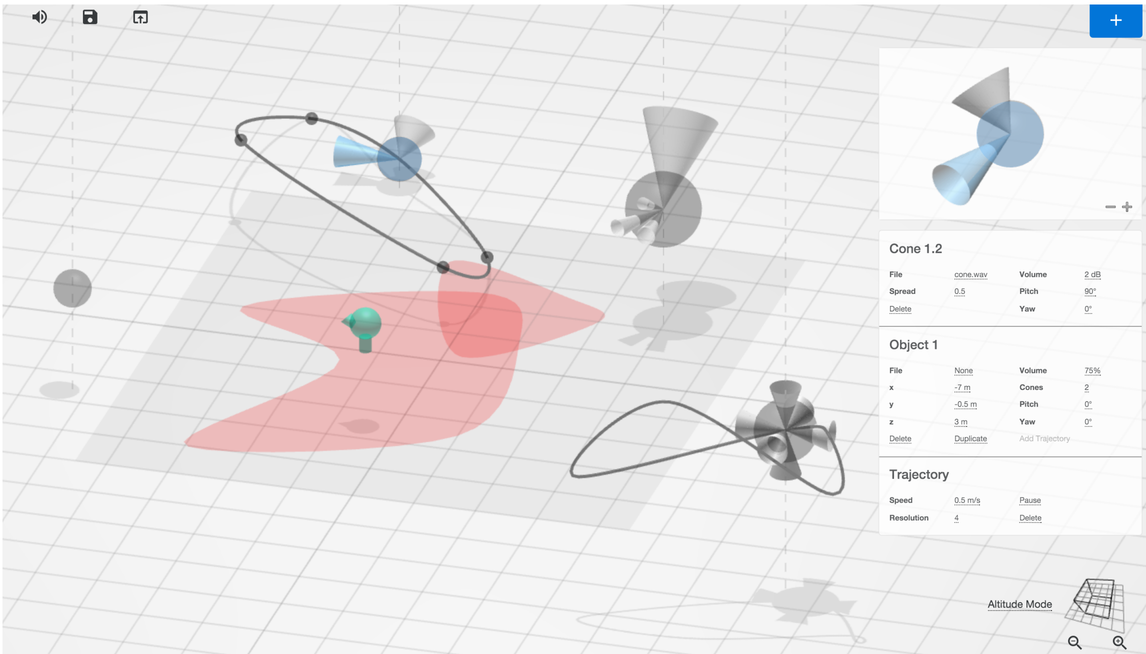

Figure 1: A screenshot of our user interface on a desktop computer displaying an object with two cones and a motion trajectory being edited. On the top right region, a close-up window displays the object with the cone that is currently being interacted with highlighted in blue. The windows below this 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.

Abstract

Current authoring interfaces for processing audio in 3D environments are limited by a lack of specialized tools for 3D audio, separate editing and rendering modes, and platform-dependency. To address these limitations, we introduce a novel web-based user interface that makes it possible to control the binaural or Ambisonic projection of a dynamic 3D auditory scene. Specifically, our interface enables a highly detailed bottom-up construction of virtual sonic environments by offering tools to populate navigable sound fields at various scales (i.e. from sound cones to 3D sound objects to sound zones). Using modern web technologies, such as WebGL and Web Audio, and adopting responsive design principles, we developed a cross-platform UI that can operate on both personal computers and tablets. This enables our system to be used for a variety of mixed reality applications, include those where users can simultaneously manipulate and experience 3D sonic environments.

*e-mail: acamci@uic.edu; web: http://anilcamci.com
†e-mail: pmurra5@uic.edu
‡e-mail: aforbes@uic.edu; web: http://www.evl.uic.edu/creativecoding/

Index Terms: H.5.1 [Information Interfaces and Presentation]: Multimedia Information Systems—Audio input/output H.5.2 [Information Interfaces and Presentation]: User Interfaces—Graphical user interfaces (GUI)

1 Introduction

A range of platforms facilitate the design of virtual environments. Most commonly, game engines, such as Unity and Unreal, are used for developing and simulating virtual realities. However, such platforms are primarily oriented towards visual design and provide only limited audio functionality. Making use of existing research into the development of interactive virtual soundscapes [6], we introduce a novel user interface 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 UI offers the following contributions:

- provides a user-friendly 3D design environment specific to sonic virtual realities, with specialized components such as sound objects and sound zones;
- offers both interactive and parametric control over the attributes of such components, enabling a precise control over

highly-detailed environments;
- 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 design virtual sound fields at various scales using multiple view and attribute windows;
- offers a unified interface for the design and the simulation of such realities, 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

The use of sound in VR dates back to the earliest implementations in this field [8]. Many studies have emphasized the role of sound in enhancing the immersive capacity of virtual environments [3, 10].

Cross-platform game engines offer basic audio functionality, such as point sources and reverberant zones. These objects are created and manipulated through the same interactions used for visual objects. Third-party developers design plug-ins to 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 limit the expressiveness of a sound designer.

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[1]. Although their tools have powerful spatializing capabilities, no user interfaces exist for creating sonic environments using them.

IRCAM's *Spat* software[2] enables the synthesis 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 unified ecosystem for virtual environment design.

2.1.1 Web Audio API

The Web Audio API [1] 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. In a project titled *Birds of a Feather*, Walker and Belet [17] used an online database of birdsong recordings in a browser based application which allows its users to synthesize dynamic soundscapes from these recordings based on the user's geolocation.

Using the Web Audio API, Rossignol et al. [14] designed an acoustic scene simulator based on the sequencing and mixing of environmental sounds on a time-line. Lastly, Pike et al. [13] 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. However, several studies have shown that non-individualized HRTFs yield inconsistent results across listeners in terms of localization accuracy [18]. Although the Web Audio API does not currently support the use of custom HRTFs, several recent studies have shown that it can be extended to allow users to upload individualized HRTFs [5, 13].

3 Overview of User Interface

A user interface for the computational design of sonic environments 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 UI approach. To represent the different elements 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 UI 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 [12]. Accordingly, in our UI, the 2D overhead 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.

We provide a unified environment for designing both sonic environments and the sound objects contained within them. We combined a multiple-scale design [2] with a dual-mode user interface [9], which improves the precision at which the user can control the various elements of the sonic environment, from sound cones to sound objects to sound fields. Local object attributes are separated from global scene controls via a secondary view with which the user can design individual sound objects.

We 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 [4], which is used in information-rich virtual environments such as ours.

Furthermore, our UI 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 taxing than graphics-based applications, editing and navigation can be performed concurrently.

Finally, we offer an amalgamation of virtual and augmented reality experiences for the user. Given the ability of our UI 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 sonic environment is superimposed, as seen in Fig. 2.

4 Sound Field

4.1 Interaction

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. With a *click&drag* action[3], the user can pan the visible area of the sound field. *Zoom* icons found on the bottom right corner allows the user to zoom in and out of the sound

[1] http://www.worldviz.com/products/vizard
[2] http://forumnet.ircam.fr/product/spat-en
[3] On mobile devices *click* is replaced by *touch* actions.

Figure 2: A user exploring the augmented reality in 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.

field. A cubic UI object found right above the zoom controls allows the user to tilt and rotate the view of the sound field.

A global mute button on the top left corner of the UI allows the user to turn off the entire audio output. This feature 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 UI states to restore a previously designed sound field.

4.2 Navigating the Virtual Sonic Environment

The user can explore the virtual sonic environment in via one of two modalities (or a combination of both of them). In *virtual navigation*, a stationary user is equipped with a headphone connected to the device running the UI. Depending on the input device, the user can either use physical or virtual arrow keys to travel within the sound field. 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 UI 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. The user is represented with a green dummy head in the scene as seen in Fig. 1.

4.2.1 Room Overlay

In augmented reality applications, 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 blue 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 [1]. 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 UI, 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* a point in the sound field to place a new sound object. The default object is an ear-level[4] omnidirectional point source represented by a translucent sphere on the sound field.

Creating a new object, or interacting with an existing object, brings up an attributes window on the top right region of the screen. On tablets, the same interaction zooms the sound field view onto the selected object, and blacks out other components on the field. In this view, the user can interact with the sound object locally and edit its attributes. On desktop computers with sufficient screen size, the same action also brings up a secondary window above the attributes window, which displays a similar close-up view of the sound object. The sound field view remains unchanged providing the user contextual control over the object that is being edited in the close-up window.

In each case, the close-up view allows the user to add or remove sound cones and position them at different pitch and yaw values. The latter is achieved by *click&dragging* a cone using an arcball interface [16]. Interacting with a cone brings up a secondary attributes window for local parameters, where the user can attach a sound file 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 attributes window also provides parametric control over pitch and yaw values.

A *Duplicate* button in the object attributes window allows the duplication of the selected object. A *Mute* button in the same window allows to turn off audio for the files attached to the selected object. 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.

5.3 Trajectories

After clicking the *Add Trajectory* button in the object attributes window, the user can *click&drag* the said object to draw a motion trajectory. If the action start and stop positions are in close proximity, the UI 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

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

depending on whether the trajectory is closed or not. Once a trajectory has been defined, a trajectory attributes 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 or internal (i.e. self-produced) sounds, we have implemented the sound zone component, which demarcates areas of non-directional and omnipresent sounds. Once the user walks into a sound zone, he or she will hear the source file attached to the zone without distance or localization cues.

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&drag* action. Once the action is completed, the UI generates a closed spline curve by interpolating between action start and stop positions. When a new zone is drawn, or after an existing zone is *clicked*, a window appears on the top right region of the screen to display zone attributes, which include audio source, volume, scale, rotation and resolution.

7 Applications

Virtual sonic environments have many applications ranging from providing assistance to people with visual impairment [15] to improving spatial perception in virtual realities [7]. The ease of use, detail of control, and the unified editing and navigation modes provided by our UI not only improve upon existing applications but also open up new practical and creative possibilities.

While our UI 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 interface in scientific applications, where researchers can rapidly construct detailed and accurate auditory scenes.

Our UI can also be used as an on-sight sketching tool by landscape architects to simulate, in 3D, 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. This would make it possible to evaluate design modifications to address issues regarding noise pollution.

Our UI opens up a variety of artistic possibilities as well. Although existing digital audio workstations allow the spatial control of sounds, the emergent spatial complexity of our sound objects would be virtually impossible to recreate with traditional interfaces. Furthermore, the real-time design features of our UI make it possible to use it as a sound performance tool.

8 Future Work and Conclusions

A next step for our UI is to include a new 3D object that enables sound occlusion. This will allow the designer to draw non-sounding objects in arbitrary shapes that affect the propagation of sounds around them. We also plan to augment the sound zones 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 facilitate rich mixed reality applications. For instance, the 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. Although our UI currently utilizes multi-touch gestures for the panning and the rotation of the sound field, we plan to incorporate further multi-touch techniques as described by Martinet et al. [11] to enhance object editing capabilities on tablets. Finally, we will investigate extending the OSC functionality of our UI to allow the control of other VR authoring tools.

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

References

[1] P. Adenot and C. Wilson. Web Audio API, 2015. [Online]. Available: http://webaudio.github.io/web-audio-api/.

[2] 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*, 7:3–31, 1995.

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

[4] 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*, VRST '03, pages 81–90, New York, NY, USA, 2003. ACM.

[5] T. Carpentier. Binaural synthesis with the web audio api. In *Proceedings of the 1st Web Audio Conference*, January 2015.

[6] A. Çamcı, Z. Özcan, and D. Pehlevan. Interactive virtual soundscapes: a research report. In *Proceedings of the 41st International Computer Music Conference*, pages 163–169, 2015.

[7] R. R. A. Faria, M. K. Zuffo, and J. A. Zuffo. Improving spatial perception through sound field simulation in vr. In *Virtual Environments, Human-Computer Interfaces and Measurement Systems, 2005. VECIMS 2005. Proceedings of the 2005 IEEE International Conference on*, pages 6–pp. IEEE, 2005.

[8] M. L. Heilig. Stereoscopic-television apparatus for individual use, October 1960. US Patent #2,955,156.

[9] 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*, WWW '12, pages 1047–1056, New York, NY, USA, 2012. ACM.

[10] M. Marchal, G. Cirio, Y. Visell, F. Fontana, S. Serafin, J. Cooperstock, and A. Lcuyer. Multimodal rendering of walking over virtual grounds. In F. Steinicke, Y. Visell, J. Campos, and A. Lcuyer, editors, *Human Walking in Virtual Environments*, pages 263–295. Springer New York, 2013.

[11] A. Martinet, G. Casiez, and L. Grisoni. The design and evaluation of 3d positioning techniques for multi-touch displays. In *3D User Interfaces (3DUI), 2010 IEEE Symposium on*, pages 115–118. IEEE, 2010.

[12] 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*, GI '05, pages 195–202, School of Computer Science, University of Waterloo, Waterloo, Ontario, Canada, 2005. Canadian Human-Computer Communications Society.

[13] 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.

[14] 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.

[15] J. Sánchez, L. Jorquera, E. Muoz, and E. Valenzuela. Virtualaurea: perception through spatialized sound. In *Proceedings of the 3rd International Conference on Disability, Virtual Reality and Associated Technology*, pages 197–204, 2002.

[16] K. Shoemake. Arcball: A user interface for specifying three-dimensional orientation using a mouse. In *Proceedings of the Conference on Graphics Interface '92*, pages 151–156, San Francisco, CA, USA, 1992. Morgan Kaufmann Publishers Inc.

[17] W. Walker and B. Belet. Birds of a feather (les oiseaux de młme plumage): Dynamic soundscapes using real-time manipulation of locally relevant birdsongs. In *Proceedings of the 1st Web Audio Conference*, January 2015.

[18] 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)*, pages 151–154, 2012.