INVISO: A Cross-platform User Interface for Creating Virtual Sonic Environments

Anıl Çamcı
University of Michigan
acamci@umich.edu

Kristine Lee
University of Illinois at Chicago
khlee2@uic.edu

Cody J. Roberts
University of Illinois at Chicago
crober29@uic.edu

Angus G. Forbes
University of California, Santa Cruz
angus@ucsc.edu

ABSTRACT

The predominant interaction paradigm of current audio spatialization tools, which are primarily geared towards expert users, imposes a design process in which users are characterized as stationary, limiting the application domain of these tools. Navigable 3D sonic virtual realities, on the other hand, can support many applications ranging from soundscape prototyping to spatial data representation. Although modern game engines provide a limited set of audio features to create such sonic environments, the interaction methods are inherited from the graphical design features of such systems, and are not specific to the auditory modality. To address such limitations, we introduce *INVISO*, a novel web-based user interface for designing and experiencing rich and dynamic sonic virtual realities. Our interface enables both novice and expert users to construct complex immersive sonic environments with 3D dynamic sound components. *INVISO* is platform-independent and facilitates a variety of mixed reality applications, such as those where users can simultaneously experience and manipulate a virtual sonic environment. In this paper, we detail the interface design considerations for our audio-specific VR tool. To evaluate the usability of *INVISO*, we conduct two user studies: The first demonstrates that our visual interface effectively facilitates the generation of creative audio environments; the second demonstrates that both expert and non-expert users are able to use our software to accurately recreate complex 3D audio scenes.

Author Keywords

Virtual sonic environments; 3D audio; virtual reality; design environment; browser-based interface.

ACM Classification Keywords

H.5.1 Information Interfaces and Presentation: Multimedia Information Systems; H.5.2 Information Interfaces and Presentation: User Interfaces

UIST 2017, October 22–25, 2017, Quebec City, QC, Canada

ISBN 978-1-4503-4981-9/17/10…$15.00

DOI: https://doi.org/10.1145/3126594.3126644

INTRODUCTION

A range of modern 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 while providing only limited audio functionality. Most modern game engines rely on a simplified implementation of spatial audio that involves omnidirectional point sources placed on a stereo panorama, and reverberant zones that help establish a sense of space. While this is sufficient for augmenting visual objects with simple sound events (e.g., footsteps attached to a human model), current desktop and mobile computers are capable of catering to designs that stem from an audio-centric perspective that would require more complex and immersive sonic environments rendered in real time.

The design of *INVISO* is based on a taxonomy of sound elements we have elaborated from an investigation of environmental sounds, soundscapes and acoustic ecology [6, 7, 8]. Accordingly, we have implemented UI elements that represent the components of this taxonomy, such as dynamic sound objects with complex propagation characteristics, and sound zones with irregular shapes. In this paper, we detail our design considerations in creating a user-friendly, cross-platform sonic VR interface, and addressing the challenges of interacting with such components individually and within the context of an auditory scene that can be populated with an arbitrary number of elements in real-time. After describing interaction scenarios for creating and manipulating auditory scenes, we offer the results of a user study conducted with *INVISO*. Finally, we provide an overview of the range of applications that our interface can support, as well as our plans to expand its capabilities.

INVISO offers the following contributions:

- **provides a 3D design environment specific to navigable sonic virtual realities, with specialized components such as sound objects and sound zones**

Our UI provides tools for the rapid design of sonic environments displayed in the form of a 3D virtual scene. Furthermore, it facilitates creating and interacting with complex spatial entities (e.g., multi-cone sound objects, sound zones with arbitrary shapes) in such scenes by adopting not only modern

Figure 1. A screenshot of our user interface on a desktop computer, displaying an *aerial view* of a virtual sonic environment. Three sound objects, which are represented with white spheres, have different cone configurations. Two of the objects follow trajectories, which can be arbitrarily drawn as open or closed loops. Three hand-drawn sound zones, two of which are overlapping, are represented with pink connected regions. The user navigates the scene by controlling the blue head model, which is currently moving away from the center of the scene. The attribute window on the right allows the user to control various parameters of the selected sound object, its cones, and its motion trajectory.

mouse and touch interactions, but also a constrained approach to 3D manipulation of sound field components, and multiple view modes for detailed control.

- **offers both interactive and parametric manipulation of sound field components at various scales, therefore catering to a wide range of users with varying degrees of experience with sound production and spatialization**

Our UI relies on common point, click, and drag interactions to draw and position sound components.[1] While these allow for the interactive manipulation of sound components, dynamic attribute windows offer numeric control over parameters. When coupled with multi-scale views for the editing of sound objects, our UI enables users with varying levels of experience with spatial audio to have precise control over highly-detailed virtual sonic environments.

- **introduces a multi-cone model for creating 3D sound objects with complex propagation characteristics**

Directionality in game audio is often achieved with sound cones, where each sound source is represented with a cone that has a unique position and orientation. However, sound producing events in nature are much more complex; a single object can produce multiple sounds with different directionality, spread, and throw characteristics. To address this, we implemented a multi-cone sound object that allows the user to attach an arbitrary number of right circular cones to individual objects, and manipulate them in 3D.

[1] On mobile devices *click* is replaced with *touch* interaction.

- **enables adding dynamism to objects via hand-drawn motion trajectories that can be edited in 3D**

Moving sound sources contribute to the articulation of space in virtual sonic environments. A 2D contour sketching interaction in *INVISO* allows users to assign spatial animation trajectories to sound objects. These trajectories can then be edited to create 3D movement.

- **offers a unified interface for the design and the simulation of virtual soundscapes, allowing the user to interact with and modify a sound field in real-time**

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 free-field audio environment that is less resource-intensive, editing and navigation is achieved in a single state, which makes the design an interactive process with real-time auditory feedback. Furthermore, given the ability of our system to function both on desktop and tablet computers, our UI offers an amalgamation of virtual and augmented reality experiences for the user. In a space with motion tracking capabilities, the user can manipulate the virtual sonic environment using a mobile device while exploring the physical space onto which this virtual environment is superimposed.

RELATED WORK

Spatial Audio Software

Various authoring tools allow the spatialization of audio. These include digital audio workstations (DAWs) and dedicated audio spatialization software. DAWs, such as Ableton Live and

Avid ProTools, are predominantly used for music production. Accordingly, the spatialization of sound in such software is achieved with rotary controllers that position sound sources on a stereo panorama, which is the dominant method of sound spatialization for commercial music. Third party VST Plug-ins, such as IOSONO AnyMix and Waves S360, extend DAWs with surround panning capabilities, which are often used in post-production for film.

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 singular interface for virtual environment design. Offering a similar interface to that of IRCAM's *Spat* suite, *Spatium* is a set of open-source software tools including standalone and plug-in application that facilitates the exploration of various spatialization techniques, and aims to integrate spatialization into "diverse computational workflows" [20]. SoundScape Renderer [11], developed by researchers at TU Berlin, is a software for the positioning of sound sources around a stationary listener using a 2D overview of the scene, where the user 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.

VR Production Software

The use of sound in virtual environments dates back to the earliest implementations of VR [12]. 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 the same interactions used for visual objects on these platforms, which are not optimized for audio processing. Additionally, third-party developers offer plug-ins, such as Steam Audio [3] and RealSpace3D,[4] 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.

As the parent company of Oculus, which develops one of the leading hardware platforms for VR, Facebook Inc. has recently published a suite of applications for immersive media production. Amongst these, *Facebook Spatial Workstation* is a DAW plug-in that allows the design of binaural sound scenes for 360 video production. Situated as a tool for "large scale consumption" of immersive media, it allows the user to position sound sources around a stationary listener and put these through various localization effects.[5]

As we discuss below, *INVISO* introduces novel functionality that enables both expert and non-expert users to design virtual sonic environments for a range of spatial audio applications.

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

[3] https://valvesoftware.github.io/steam-audio/

[4] http://realspace3daudio.com

[5] https://medium.com/@katya_alexander/designing-the-sound-of-reality-680729bdae7a

Virtual Acoustic Environments

Studies on virtual acoustic environments (VAEs) investigate the modeling of sound propagation in virtual environments through source, transmission, and listener modeling [25]. In the 1990s, Huopanemi et al. [13] developed DIVA Virtual Audio Reality System as a real-time virtual 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 treatment purposes. In such applications, specialized software allows 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 [9, 30]. While previous projects have offered efficient methods for the rendering of virtual acoustic environments [17, 15, 31], 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 [15]. In order to make cross-platform operation possible in the browser, *INVISO* avoids the complexities of acoustic modeling by focusing on free-field environments, introducing novel sonic objects that adequately define a virtual scene.

Applications

Chris Schmandt's *Audio Hallway* project offers a virtual acoustic environment for browsing collections of audio files. As the users virtually navigate through the *Audio Hallway*, spatial audio guides them into rooms, where they can hear sound bites related to broadcast news stories [26]. Mynatt et al.'s *Audio Aura* utilizes augmented reality audio to provide office workers with auditory cues, such as information about their coworkers, emails or appointments, as they walk through the office [18]. Similarly, in their *Hear&There* project, Rozier et al. propose an augmented audio system where users can place geo-located "audio imprints," which can then be heard by others who visit the imprinted locations [24]. In an early implementation of virtual sound spaces, the *ENO* system offered a Unix environment for the design of such spaces using high-level descriptors such as sound sources, their attributes and interactions. The system is aimed at facilitating the creation of "consistent, rich systems of audio cues" [2]. Using the Web Audio API, Rossignol et al. [23] designed an acoustic scene simulator based on the sequencing and mixing of environmental sounds on a time-line. Lastly, Pike et al. [21] 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. The latter examples demonstrate that Web Audio, which our UI relies on for sound rendering, is powerful enough to support the design of sonic virtual realities in the browser.

SYSTEM DESIGN

To enable operation over various hardware and software platforms, *INVISO* has been designed for the web-browser using

modern audio and visual rendering APIs. This approach not only enables a platform-independent extensibility of our system, but also opens up a variety of virtual and augmented reality applications through its use on desktop and mobile computing devices.

For the rendering of visuals in *INVISO*, we use the JavaScript WebGL library *three.js*,[6] while the graphical user interface elements are implemented with CSS. For the calculation of the centripetal Catmull-Rom splines, which make up zones and trajectories, we use the polygon tessellation library *libtess*.[7]

For the rendering of the audio in *INVISO*, we use the Web Audio API [1], which is a JavaScript library for processing audio in web applications. Based on the idea of *audio graphs*, Web Audio API offers abstractions for creating sound spatializer and listener nodes, which help modularize the auditory components of our UI similarly to its visual elements. Our system utilizes the built-in binaural functionality of the Web Audio API, which is derived from *IRCAM Listen*'s head-related transfer function (HRTF) database [5].

UI CONSIDERATIONS

A Constrained Approach to a 3D UI

Our UI utilizes a 3D visual scene, which the user can view at different angles in order to add and edit different sound 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 more effective than single-view controls with axis handles [19]. Accordingly, in our UI, we rely on two primary view modes that are activated based on the camera angle controlled by the user (see Fig. 2(a) and (b)): when the user is looking at the scene from a bird's-eye view, the scene is in the *aerial view*, which enables the movement of sound components on the horizontal plane. When the user tilts the camera to change the perspective, the scene switches to the *altitude view*, which enables the movement of sound objects and trajectory points on a vertical axis. When in the *altitude view*, if the user clicks the view label, the scene immediately switches back to the *aerial view*.

Dual-mode UI for Contextual Interactions

We provide a unified environment for designing both sonic environments and the sound objects contained within them. To achieve this, we combined a multiple-scale design [3] with a dual-mode user interface [14], 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. When the user is controlling the spatial distribution of the objects, they do so within the main mode of the interface, which affords an either *aerial* or *altitude view* of the sound field. On the other hand, when the user wants to alter the internal structure of a sound object, the interface switches to an *object-edit mode* as seen in Fig. 2(c), which zooms into the selected object in the scene, and constrains the interactions to the cones of this object. In this mode the camera orbits around the object that is locked in its place, giving the user to ability to listen to the object from different orbital points.

Dynamic Windows for Detailed Control

We utilized dynamic attribute windows, or inspectors, to offer parametric control over properties that are normally controlled via mouse or touch interactions. This enables a two-way communication between abstract properties (i.e. parameters) and the virtual environment in a combined design space [4], which is used in information-rich virtual environments such as ours: when the user moves an object in the scene interactively, an attribute window for this object appears as seen in Fig. 1; the information displayed in this window gets updates upon the user's interaction with the object. In this case, the user receives a precise feedback about the object's parameters. Conversely, the user can enter explicit parameter values, which in return manipulate the object in the virtual environment, offering a precise control over the design of an object.

A VISUAL TAXONOMY OF SPATIAL AUDIO

A user interface for the computational design of sound requires audio-to-visual representations. In DAWs, audio tracks are represented by horizontal strips that extend over a timeline, where the user can edit sound files by splicing portions of this strip. Furthermore, multiple tracks can be aligned vertically to create simultaneous sound elements. However, in the context of a virtual reality application, sound elements need to be conceived primarily as spatial entities, which require a different UI approach than what modern DAWs offer. Previous work by Çamcı et al. [8] investigates a taxonomy of components that make up a sonic VR based on the directionality characteristics of sound sources. The resulting classification includes uni-directional, omni-directional, and non-directional sounds. In the current study, we investigated the geometries that can be used to represent these components, and the visual interaction methods necessary to create and manipulate these geometries in 3D.

Uni-directional Sounds: Cones

Uni-directional sounds are those that propagate in a limited area of projection. For instance, when we speak, our voice propagates in the direction of our mouth, giving our audience a sense of localization and focus. Sound producing or amplifying apparatus, such as loudspeakers and megaphones, rely on a similar principal, also known as *waveguiding*. Most acoustic waveguiding systems utilize a horn model to achieve directionality: a cone found in a loudspeaker, for instance, serves to focus its sound projection in a specific direction, while where the loudspeaker itself is located gives a sense of localization for the point from which the sound propagates.

In virtual systems, such as games, a similar behavior is implemented with *sound cones* [1]. In such systems, a virtual uni-directional sound source is often represented with two nested cones: 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 defines an extended region in which the user hears an attenuated version of the same file. This allows a directional sound object to fade in and out of the audible space.

[6] https://threejs.org
[7] https://www.npmjs.com/package/libtess

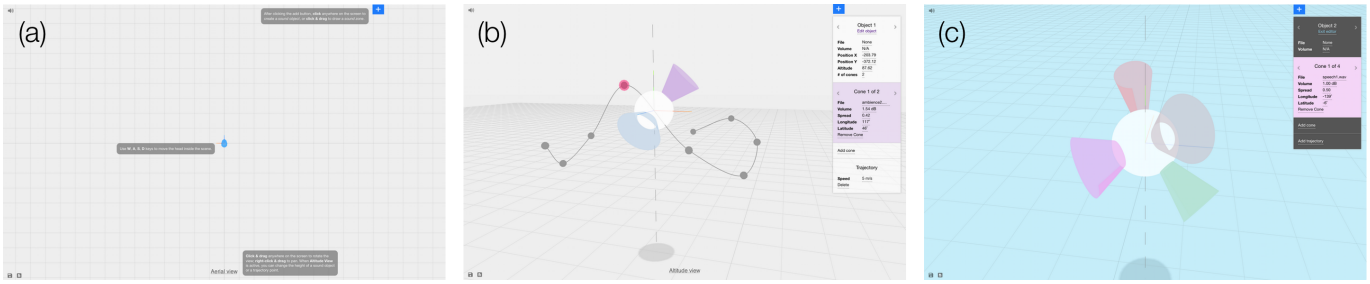

Figure 2. (a) An image of the system on start-up, displaying an *aerial view* of the scene with three help bubbles that are sufficient to get a novice user to start designing a virtual sonic environment. The simplified interface offers a single point of entry, a blue *plus button*, to create new sonic components. In *aerial view*, the user can position sound components on the horizontal plane. (b) In this image, the user is editing the 3D motion trajectory of an object in *altitude view*, while having access to the objects parameters through the attribute window to the right on the screen. In *altitude view*, the user can manipulate an object's position on a vertical axis. (c) This image shows the *object-edit mode* where the user has both interactive and parametric control over the finer details of a sound object including cone position, spread and volume.

Omni-directional Sounds: Spheres

An omni-directional sound is similar to uni-directional sound in that it is a point source localized in the audible space; however, instead of a limited conic projection, the omni-directional sound has a spherical propagation characteristic.

While all sounds in nature are directional to a certain extent based on the physical phenomenon that causes the changes in air pressure, some events, such as two hands clapping or a stick breaking, have wider-angle projections compared to that of waveguided sound sources. In sound reproduction, omni-directional speakers offer an array of cones aligned to the surfaces of a regular polyhedron, such as a dodecahedron, to emulate the behavior of omni-directional sound sources.

As an omni-directional sound moves in space, even though its localization characteristics change, its directionality characteristics remain the same in reference to the listener; in that sense, an omni-directional sound can be envisioned as a uni-directional sound source that is always directed at the listener. In music, film, and games, such sounds are the most commonly used elements of spatial audio: as a sound is moved on the stereo panorama, for instance, it still remains directed at the origin (i.e. the "sweet spot") where the listener is assumed to be.

Non-directional Sounds: Simply Connected Regions

Non-directional sounds correspond to ambient sounds in the environment that does not incite a clear sense of localization. Examples to these can be wind, traffic, and crowd sounds, where the listener is immersed by the sounds without pinpointing individual localized sources. Rather than a single point of origin, such sounds can be attributed to an acoustic zone. Once in this zone, the user is surrounded by the non-directional sound. Multiple zones can overlap; furthermore, a zone can contain a number of omni- or uni-directional sounds. Non-directional sound zones can be represented with 2D simply connected regions.

COMPONENTS OF INVISO

Sound Field

On start-up, the UI offers an overhead view of the sound field articulated with a 2D grid as seen in Fig. 2(a). This grid represents the area onto which the user can place sound components in order to create a virtual sonic environment. The grid also demarcates the plane that the user can navigate virtually. Using virtual joysticks on a tablet or W, A, S, D keys on a keyboard, the user moves a blue head model in the scene. A Web Audio *ListenerNode* attached to this model allows the user to hear a binaural rendering of the sound field through headphones based on the location and the orientation of the head model.

A blue *plus* button represents the first point of entry to designing a scene. Once this button has been clicked, clicking or click-and-dragging anywhere on the sound field creates a new sound component. Once populated with sound components, the sound field can be saved in order to retain all the elements of the scene, including the head-model position and the camera orientation. Saved sound fields can later be loaded. Saving and loading is achieved with individual buttons on the bottom left side of the screen. A global *mute* button on the top left corner of the UI allows the user to turn off the entire audio output and pause the spatial animation of objects. This feature makes it possible to make offline modifications to the sound field.

By click-and-dragging any unpopulated region on the screen, the user can rotate the perspective at which the sound field is viewed. Besides offering a sense of 3D context, this action allows the user to switch between *aerial* and *altitude views* when a certain threshold of rotation is crossed. In *aerial view*, the user can move all the sound components in the scene on the horizontal plane. If the user rotates the polar angle of the camera more than 20 degrees, it switches to *altitude view*, as indicated by a label on the bottom middle of the screen. In this mode, the user can change the vertical position of sound objects and trajectory control points, which are now locked in place on the horizontal plane. When in *altitude view*, clicking a button to either add a new sound component or a motion trajectory switches the scene to *aerial view* to allow for the placement or drawing of new components.

Secondary click-and-dragging allows the sound field to be panned while two-finger scrolling zooms in and out of the sound field, offering either a more detailed or a broader view of the scene.

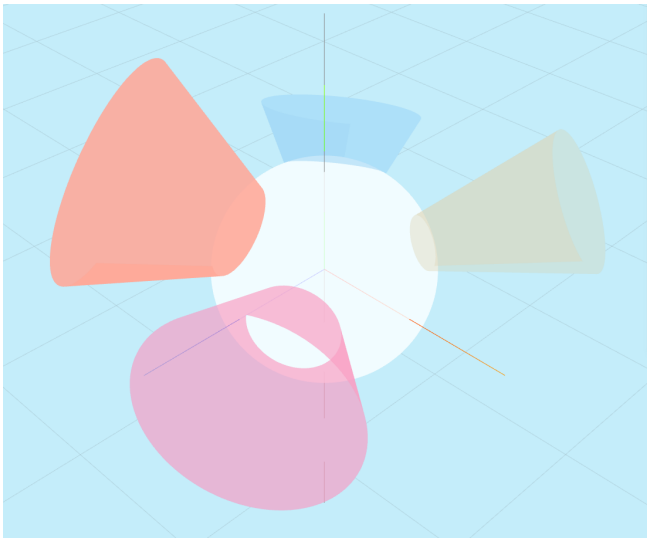

Figure 3. A screenshot of the *object-edit mode* displaying a sound object with four cones.

Sound objects

Multi-cone implementation

A sound producing object or event in an everyday soundscape can have multiple sounds emanating with diverse directionality, throw, and spread characteristics. To simulate such phenomena, we improved upon the common sound cone model, and implemented a multi-cone sound object, as seen in Fig. 3, that allows the user to attach an arbitrary number of right-circular cones to individual objects, and adjust them in 3D.

Interaction

After clicking the *plus* button on the top right corner of the UI, the user can then click anywhere in the sound field to place a new sound object. The default object is an ear-level[8] omni-directional point source represented by a translucent sphere on the sound field.

Creating a new object, or interacting with an existing object, brings up an attribute window, as seen in Fig. 1, that provides precise control over the object's parameters.

The *object-edit mode*, seen in Fig. 2(c), allows the user to add or remove sound cones and position them at different longitude and latitude values. The latter is achieved by click-and-dragging a cone using an arcball interface [29]. Interacting with a cone brings up a secondary attribute window for local parameters, where the user can replace the sound file attached 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. This window also provides parametric control over longitude and latitude values.

Trajectories

After clicking the *Add Trajectory* button in an object's attribute window, the user can click-and-drag the said object to draw a motion trajectory. If the start and stop positions are in close proximity, the UI interpolates between these points to form a closed loop. Once the action is completed, the object will begin to loop this trajectory with either back-and-forth or circular motion depending on whether the trajectory is closed or not. Using the trajectory speed attribute, the user can speed up, slow down, stop, or reverse the motion of the object. Once the user clicks on an object or its trajectory, control points that define the trajectory's spline curve become visible. While the lateral position of these positions can be altered in the *aerial view*, the *altitude view* allows the user to change the height of individual points, therefore allowing the design of 3D motion trajectories as seen in Fig. 2(b). Furthermore, the user can add or delete control points to alter an existing trajectory.

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

Sound Zones

To enable the user to create ambient sound sources, 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 hear the source file attached to the zone without distance or localization cues.

Interaction

After clicking the blue *plus* button, the user can click and drag within the sound field to draw a zone of arbitrary size and shape. Once the action is completed, the UI generates a simply connected region, as seen in Fig. 4, by interpolating between the action's start and stop positions. When a new zone is drawn, or after an existing zone is selected, an attribute window appears on the top right, offering parametric control over the zone's audio file, volume, position and rotation. Multiple zones can be overlapped to create complex ambient sound structures.

COMPARISON TO OTHER INTERFACES

Primary applications of spatial audio include music and film production. As a result, most audio spatialization tools assume a stationary user at the center of a spatial audio scene, such as those created for surround sound in films, or stereo mixes in music. This is emphasized in most sound spatialization tools,

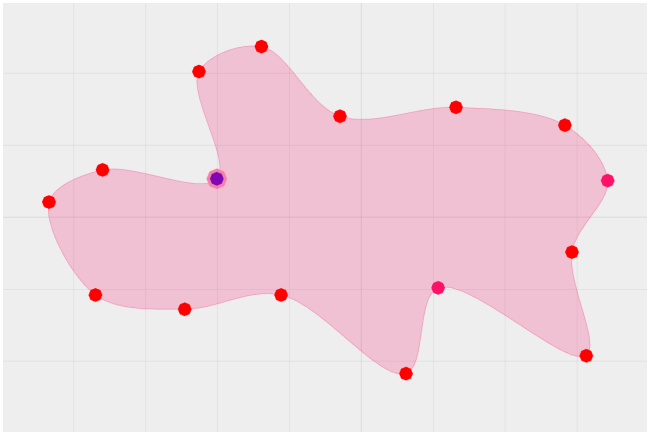

Figure 4. A screenshot of a sound zone that is currently being edited. The user is adding a new control point at the location highlighted with the blue dot.

such as *Spatium* and *Spatial Audio Workstation*, with an interaction paradigm where a scene is populated with point sources around a central listener node. While one of the modules in IRCAM's *Spat* software allows the listener node to be moved on a 2D plane, this module is part of a comprehensive set of sound spatialization tools mainly intended for specialists who work with spatial audio.

Most digital audio workstations (DAWs) are designed to emphasize the temporal progression of music rather than its spatial layout to allow for precise time editing of sound files. These software systems also offer means to position sounds on a stereo panorama using 2D sliders, which are sufficient to address the needs of modern music production where most of the releases are in stereo format. Using multiple tracks, these 2D UI elements can be used to create surround sound diffusion. However, manipulation of multiple sound sources quickly becomes cumbersome, as the user is forced to conceive a spatial scene in terms of a combination of 2D (i.e., value-time) UI automations.

The existing UI standards for controlling spatial audio in VR authoring tools are inherited from the visual design paradigm, and are overly cluttered for manipulating audio. For instance, to design a sound field in Unity, the user would need to create a 3D object, add an audio source node to this object, add assets to the project, bind these assets to the source node, and adjust its parameters for 3D diffusion. The user would then need to create a camera or a third person controller and attach an audio listener node to it. Given the existing audio features of Unity, the resulting environment would consist of basic point sources and reverberant zones.

In comparison, *INVISO* was designed to adopt the navigable environment features of VR authoring tools, but offers audio-specific features that will enable both expert and non-expert users to design such environments for spatial audio applications. To achieve this, our UI relies on a simple set of features, which can be combined and parametrically controlled to create complex sonic environments. In that sense, we believe that our UI follows Shneiderman's design principle for creativity support tools–it has a low barrier of entry for novice users, while yet it offers high ceilings for experts [28].

USER STUDY

To evaluate the usability of our UI, we have conducted an exploratory user study, which included two tasks.

Method

Participants

10 users with varying degrees of experience with digital audio workstations, sound spatialization software, and VR authoring platforms were asked to participate in the user study. Fig. 5 shows the distribution of the participants' level of experience in these areas.

Procedure

The study was administered one user at a time. Each study was administered on a desktop computer equipped with closed-back headphones. All the participants were given a 5-minute verbal description of the application and the format of the user study. The entire study took 30 minutes on average.

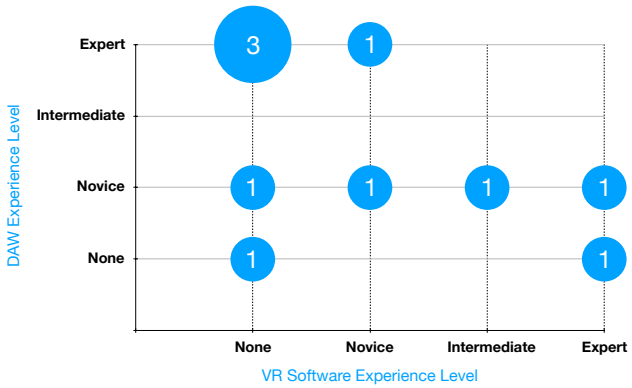

Figure 5. Participants' experience levels with VR authoring software (e.g., Unity) on the X-axis, and digital audio workstations (e.g., Avid Pro-Tools) on the Y-axis. Numbers in the bubbles represent number of users who selected a particular experience level combination.

The study involved two tasks. In both tasks, the participants were given a set of 10 audio files, which included ambient, environmental, noise, music, and speech sounds. The first task, where we asked the participants to design an arbitrary sound scene without a particular instruction or constraint, aimed to gauge the usability of the interface for first-time users with varying degrees of expertise in sound and virtual reality production. In the second task, *INVISO* was loaded with a pre-defined scene with the visual representations of its sound components made invisible. The participants were asked to navigate and listen to this scene, and to roughly recreate it on a separate instance of the application. The participants were told that they could switch between the pre-defined scene and their recreation of it. They were also told that all the invisible components in the pre-defined scene were contained within the current zoom-level of the sound field, and that the participants did not have to zoom out, pan, or navigate outside the visible area of the sound field. This second portion was aimed at investigating the repeatability of scenes, the ease of auditory interaction with the system, and the accuracy of the binaural rendering of the scene. A time constraint was not been imposed in either section of the study.

After the completion of the second section, the users were asked to complete a survey, which included multiple-choice and written questions about their experience with the interface. Finally, the users were given the opportunity to expand upon their views about the interface in the form of an informal discussion.

Results and Discussion

The participants spent 10.6 minutes on average to complete the first task, and 12.63 minutes on average to complete the second task. A box plot of completion times for each task as seen in Fig. 6 shows the range of durations and the median task times across users.

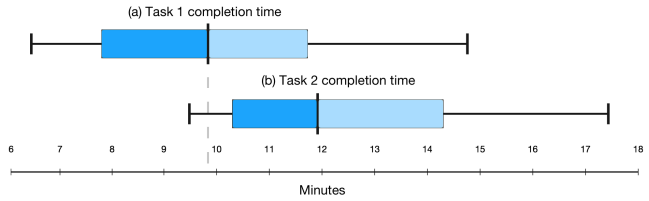

Figure 6. Box plot of times spent for the completion of (a) first and (b) second tasks.

Task 1

Fig. 7 shows screen captures from the virtual sonic environments designed by 6 of the 10 participants during the first section of the study. These images display various creative approaches the participants were able to apply to their designs. All users except for the designer of Fig. 7(c) had no or novice-level experience with DAWs and audio spatialization. This distinction is apparent in Fig. 7(c) in that the participant chose a more traditional approach to sound spatialization with a stationary user in the center with sound sources placed around this center at radial positions.

Reflecting upon his design during the informal discussion that followed up the study, the designer of Fig. 7(a) described that the 3D motion trajectories has inspired him to create a railway-like system. While Fig. 7(b) shows a map of sound islands created with zones, Fig. 7(d) follows a similar approach in creating a topographic map of sound zones. While the designer of Fig. 7(e) focused on generating complex sound objects with a dense array cones, the designer of Fig. 7(f) explored a complex spatial arrangement that included spiraling motion trajectories. This user described that his goal was to create rhythmic structures out of spatial motion of sound objects.

Regardless of their levels of expertise with spatial audio or virtual reality, all users were able to create scenes with multiple sound zones and objects with cones and motion trajectories. Given that all of the participants were first-time users, the participants' ability to create new virtual sonic environments within minutes, and the diversity of creative approaches in the design of these environments are encouraging outcomes of the first task, in terms of the usability of our system. Administering an open-ended creative task has also prompted the participants to propose new features and applications, which we will discuss below.

Task 2

The second section of the user study, where the users were asked to recreate a sonic environment using *INVISO*, has highlighted several aspects of the system. Fig. 8(a) shows the scene provided to the participants with invisible objects, while (b) to (h) shows a selection of the participants' recreation of this scene. When given the task to roughly recreate a scene, the participants were successful in pinpointing sound zones and stationary sound objects. While there is a variation in how a moving sound object was interpreted in terms of the shape of its motion trajectory, this variation can be due to the perceptual similarities between the audible outcome of circular and linear trajectories. This confusion is primarily due to the

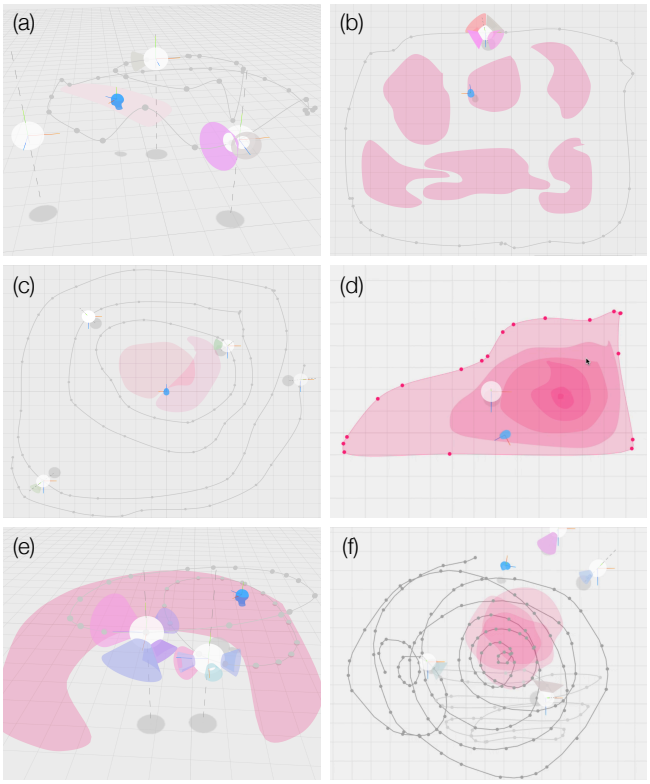

Figure 7. A selection of screen captures from the first task indicate the range of virtual sonic realities that were created by participants.

poor back-and-forth imaging inherent to non-individualized HRTFs [32], such as the one used in Web Audio API.

Only 1 in 10 users were able to correctly recreate the top left sound object with 2 orthogonal cones. While most users were able to map one of the cones correctly, the second cone was allocated to either a cone on another object, or an omni-directional sound object itself, with perceptual outcomes similar to that of a multi-cone object. While this result indicates that further experience with the multi-cone model might be necessary to better evaluate such objects, it also implies a flexibility in the UI that allows the creation of a specific spatial sound idea in various ways.

The results of this section informs us about the participants' ability to reason about spatial audio in terms of the visual representations provided in *INVISO*. A surprising result of this task was that the semblance of a user's rough approximation to the original design was not noticeably impacted by the user's experience with spatial audio and virtual reality. As a result, this task provides us with preliminary evidence that contradicts the assumption of an expert user achieving a better mapping between virtual space and sound, indicating that our UI provides appropriate functionality to both expert and non-expert users for designing virtual soundscapes.

Qualitative Feedback

As shown in Fig. 9, all participants reported a high correlation between the spatial sounds they heard when using the UI and their visual representations. The mapping between different

Figure 8. (a) The scene provided to the users with invisible objects in the second section of the study; (b)-(h) Recreations of this scene by 7 of the participants.

types of spatial sounds (i.e. non-directional, omni-directional, uni-directional) and the visual geometries (i.e. simply connected regions, spheres, cones), was described as being clear and straightforward. For instance, one participant with no experience with sound spatialization software reported that he did not need to spend time to understand how spatial audio concepts mapped to visual objects, and he was immediately able to think through his design in terms of a sonic environment rather than a visual one. Another user, also without any experience using audio spatialization software, similarly described that it was not hard to decide which types of objects to use in order to realize an idea.

Some participants expressed that the differentiation between the *aerial* and *altitude views* were not immediately clear to them but as they worked with the UI, they became accustomed to it. Some participants described the *object-edit mode* as one of the more complicated features of the UI. However, on a Likert scale between "complicated" (0) and "straightforward" (5), the users rated the 3D interactions with a sound object a 4.4 on average. Two participants expressed a need to be able to duplicate cones and objects.

A feature requested by two other participants was the ability to create radial gradient sound zones that allow the sound of a zone to fade in as a user walks into it. While the current zone implementation offers brief fade-ins and outs to avoid abrupt

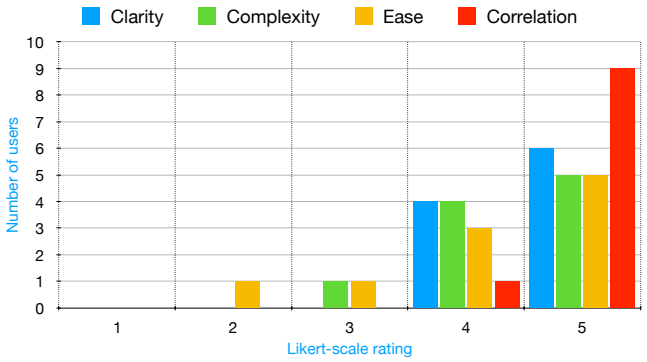

Figure 9. In four Likert-scale questions the participants rated a) clarity and accessibility of the features of the UI (1: not clear at all, 5: very clear); b) complexity of creating and manipulating 3D sound objects with the UI (1: complex, 5: straightforward); c) ease of realizing a spatial sound idea with the UI (1: hard, 5: easy); d) correlation between visual objects and sounds (1: not correlated at all, 5: highly correlated).

changes in audio, the users do not have control over the extent of these transitions. Another feature which was requested by 3 participants was the ability draw trajectories to create dynamic sound zones. 6 participants described the trajectory drawing as their favorite part of the UI. A participant expressed wanting to rotate trajectories similar to how this can be done with the zones.

Overall, participants often used such descriptors as "intuitive," "fun," and "easy to use" to describe various features of the UI. 2 users reported that their experience with gaming helped them when navigating a sound scene during the tasks. A participant with expert understanding of VR authoring tools described that "placing objects and other environmental factors in Unity" was similar to spatializing sounds in *INVISO*. Another participant with a similar level of experience with game engines expressed that omni-directional sound objects were similar to audio elements in game engines, but the remaining features of *INVISO* "don't really exist" in these platforms.

When comparing digital audio workstations with *INVISO*, a participant likened spatializing audio with *INVISO* to a game, and said that seeing spatial audio concepts visualized in 3D was helpful. A participant with expert understanding of DAWs stated that it could be possible to create similar scenes with DAWs but there wouldn't be an immediate connection between the sound event they have in mind and the work that would need to be put into creating it with a DAW. Finally, another similarly experienced user expressed that *INVISO* could be very useful in communicating ideas about soundscapes and sound installations, where spatial navigation is a crucial part of the experience. He added that sharing an *INVISO* export would be a much more intuitive method of describing such applications as opposed to doing so in written text.

APPLICATIONS

The ease of use, the 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. Most basically, *INVISO* can function as a sound field sketching tool that scales to consumer

grade computers. This allows the use of our UI as a creativity support tool in a range of applications. Artists can utilize *INVISO* to create spatial compositions and sound installations. Based on the results of our user study, we also see a potential use of our UI as a casual creativity tool, akin to graphical editors, such as Microsoft Paint and Adobe Photoshop, which are often used by non-professionals. A consumer can, for instance, easily create dynamic musical or theatrical experiences by loading music, speech or environmental sound files as moving or stationary components in binaural space.

INVISO can also serve as an auditory display tool. Shinn-Cunningham lists four main areas of application for virtual auditory displays as scientific studies of sound localization, clinical evaluation, display of spatial information, and entertainment [27]. Furthermore, McMullen proposes that auditory virtual worlds can be used to reduce cognitive load and improve performance during data analysis, to increase the feeling of immersion in training simulations, to improve the plausibility of virtual environments that are used to aid people with social anxiety, and to introduce people with visual impairments to virtual reality [16].

We believe that *INVISO* can function as a flexible and powerful platform in all these areas. The role of audio signals in providing three-dimensional or out-of-sight information has been studied by [10, 16]. *INVISO* can be used in immersive information and scientific visualization environments, such as CAVEs [22], to direct the viewer's attention to regions of significance, which the viewer's gaze is directed away from. Given its ability to be dynamically populated, our UI can also be used to simulate environments with a high density of aural stimulants in order to investigate the effects of information fatigue on cognitive performance.

The parametric control features of our UI allows it to be utilized as a sonification interface in scientific applications, where researchers can rapidly construct detailed and accurate auditory scenes. Furthermore, since the browser can receive and transmit Open Sound Control[9] (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 platforms with OSC capabilities, such as Processing, openFrameworks, and Unity. This way our interface can be used as a front end to various other software and hardware platforms, including surround sound systems.

Our system can also be used as an on-sight sketching tool by landscape architects to simulate the sonic characteristics of open-air environments in 3D. By mapping a physical environment on the virtual sound field, the architect can easily simulate a sonic environment.

FUTURE WORK AND CONCLUSION

While recent developments in virtual reality technologies are largely focused on the visual domain, we believe that virtual audio is an increasingly significant aspect of designing immersive environments. In addition to addressing feature requests by the participants of our user study, such as gradient sound zones, zone trajectories, and object duplication, we plan to implement a series of new futures which will expand the application domain of our UI. For instance, we will add the ability to attach audio streams, as opposed to files, to sound components, which will enable the real time virtualization of everyday sonic environments. Another imminent feature we plan to implement is the creation of 3D objects that enable sound occlusion, which will allow the designer to draw non-sounding objects in arbitrary shapes that affect the propagation of sounds around them. Using the hardware features of modern mobile devices, we can facilitate rich mixed reality applications with our UI. For instance, incorporating the video stream from a tablet's rear-facing 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. Paired with GPS data from such mobile devices, large-scale environments can be populated with virtual sound components. Finally, an area we foresee great potential for our system is online, multi-user collaboration. The current saving and loading mechanisms are designed to support this functionality, where instead of requiring the user to explicitly save a sound field, asynchronous state saving and sharing will enable remote collaborations inside a single audio scene.

In this paper, we introduced *INVISO*, a novel system to create and interact with sonic virtual realities. Our system provides an easy-to-use environment to construct highly-detailed, immersive 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 design complex sound objects and sound zones with dynamic attributes, parametric controls, and multiple viewports to simplify 3D navigation. Two user studies demonstrate that both expert and non-expert users are able to create and re-create complex audio scenes using the different views and interaction features provided by *INVISO*, indicating that it provides new creative and practical possibilities for designing and experiencing sonic virtual environments.

[9]Open Sound Control is a cross-platform networking protocol for multimedia systems. http://opensoundcontrol.org

REFERENCES

1. Paul Adenot and Chris Wilson. 2015. Web Audio API. (2015). Available: http://webaudio.github.io/web-audio-api/.
2. Michel Beaudouin-Lafon and William W Gaver. 1994. ENO: synthesizing structured sound spaces. In *Proceedings of the 7th Annual ACM Symposium on User Interface Software and Technology (UIST)*. 49–57.
3. Benjamin B. Bederson, James D. Hollan, Ken Perlin, Jonatham Meyer, David Bacon, and George Furnas. 1995. PAD++: A Zoomable Graphical Sketchpad for Exploring Alternate Interface Physics. *Journal of Visual Languages and Computing* 7 (1995), 3–31.
4. Doug A. Bowman, Chris North, Jian Chen, Nicholas F. Polys, Pardha S. Pyla, and Umur Yilmaz. 2003. Information-rich Virtual Environments: Theory, Tools, and Research Agenda. In *Proceedings of the ACM Symposium on Virtual Reality Software and Technology (VRST)*. 81–90.

5. Thibaut Carpentier. 2015. Binaural synthesis with the Web Audio API. In *Proceedings of the 1st Web Audio Conference*.
6. Anıl Çamcı, Paul Murray, and Angus G. Forbes. 2016a. Designing and Controlling Virtual Sonic Environments Using a Browser-based 3DUI. In *Proceedings of the IEEE Symposium on 3D User Interfaces (3DUI)*. 235–236.
7. Anıl Çamcı, Paul Murray, and Angus G. Forbes. 2016b. A Web-based System for Designing Interactive Virtual Soundscapes. In *Proceedings of the 42nd International Computer Music Conference (ICMC)*. 579–584.
8. Anıl Çamcı, Zeynep Özcan, and Damla Pehlevan. 2015. Interactive Virtual Soundscapes: a Research Report. In *Proceedings of the 41st International Computer Music Conference (ICMC)*. 163–169.
9. Thomas Funkhouser, Jean Marc Jot, and Nicolas Tsingos. 2002. "Sounds good to me!"–Computational sound for graphics, virtual reality, and interactive systems. *ACM SIGGRAPH Course Notes* (2002), 1–43.
10. Bo Gehring. 1988. Attitude Indicator. (27 September 1988). US Patent #4,774,515.
11. Matthias Geier and Sascha Spors. 2012. Spatial Audio with the SoundScape Renderer. In *27th Tonmeistertagung–VDT International Convention*.
12. Morton L. Heilig. 1960. Stereoscopic-television apparatus for individual use. (4 October 1960). US Patent #2,955,156.
13. Jyri Huopaniemi, Lauri Savioja, and Tapio Takala. 1996. DIVA virtual audio reality system. In *Proceedings of International Conference on Auditory Display (ICAD)*.
14. Jacek Jankowski and Stefan Decker. 2012. 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)*. 1047–1056.
15. Mikko-Ville Laitinen, Tapani Pihlajamäki, Cumhur Erkut, and Ville Pulkki. 2012. Parametric time-frequency representation of spatial sound in virtual worlds. *ACM Transactions on Applied Perception (TAP)* 9, 2 (2012), 8:1–20.
16. Kyla McMullen. 2014. The potentials for spatial audio to convey information in virtual environments. In *IEEE VR Workshop on Sonic Interaction in Virtual Environments (SIVE)*. 31–34.
17. Ravish Mehra, Atul Rungta, Abhinav Golas, Ming Lin, and Dinesh Manocha. 2015. WAVE: Interactive Wave-based Sound Propagation for Virtual Environments. *IEEE Transactions on Visualization and Computer Graphics* 21, 4 (2015), 434–442.
18. Elizabeth D Mynatt, Maribeth Back, Roy Want, and Ron Frederick. 1997. Audio Aura: Light-weight audio augmented reality. In *Proceedings of the 10th Annual ACM Symposium on User interface Software and Technology (UIST)*. 211–212.
19. Ji-Young Oh and Wolfgang Stuerzlinger. 2005. Moving Objects with 2D Input Devices in CAD Systems and Desktop Virtual Environments. In *Proceedings of Graphics Interface*. 195–202.
20. Rui Penha and J Oliveira. 2013. Spatium, tools for sound spatialization. In *Proceedings of the Sound and Music Computing Conference*. 660–667.
21. Chris Pike, Peter Taylour, and Frank Melchior. 2015. Delivering Object-Based 3D Audio Using The Web Audio API And The Audio Definition Model. In *Proceedings of the 1st Web Audio Conference*.
22. Khairi Reda, Alessandro Febretti, Aaron Knoll, Jillian Aurisano, Jason Leigh, Andrew Johnson, Michael E Papka, and Mark Hereld. 2013. Visualizing large, heterogeneous data in hybrid-reality environments. *IEEE Computer Graphics and Applications* 4 (2013), 38–48.
23. Mathias Rossignol, Gregoire Lafay, Mathieu Lagrange, and Nicolas Misdarris. 2015. SimScene: A web-based acoustic scenes simulator. In *Proceedings of the 1st Web Audio Conference*.
24. Joseph Rozier, Karrie Karahalios, and Judith Donath. 2000. Hear&There: An Augmented Reality System of Linked Audio. In *Proceedings of International Conference on Auditory Display (ICAD)*.
25. Lauri Savioja, Jyri Huopaniemi, Tapio Lokki, and Ritta Väänänen. 1999. Creating Interactive Virtual Acoustic Environments. *J. Audio Eng. Soc* 47, 9 (1999), 675–705.
26. Chris Schmandt. 1998. Audio hallway: A virtual acoustic environment for browsing. In *Proceedings of the 11th Annual ACM Symposium on User Interface Software and Technology (UIST)*. 163–170.
27. B Shinn-Cunningham. 1998. Applications of virtual auditory displays. In *Proceedings of the International Conference of the IEEE Engineering in Biology and Medicine Society*, Vol. 3. 1105–1108.
28. Ben Shneiderman. 2007. Creativity support tools: Accelerating discovery and innovation. *Commun. ACM* 50, 12 (2007), 20–32.
29. Ken Shoemake. 1992. ARCBALL: A User Interface for Specifying Three-dimensional Orientation Using a Mouse. In *Proceedings of the Conference on Graphics Interface*. 151–156.
30. Tapio Takala and James Hahn. 1992. Sound rendering. *Computer Graphics* 26, 2 (1992), 211–220.
31. Tan Yiyu, Yasushi Inoguchi, Eiko Sugawara, Makoto Otani, Yukio Iwaya, Yukinori Sato, Hiroshi Matsuoka, and Takao Tsuchiya. 2011. A real-time sound field renderer based on digital Huygens' model. *Journal of Sound and Vibration* 330, 17 (2011), 4302–4312.
32. Shengkui Zhao, Ryan Rogowski, Reece Johnson, and Douglas L Jones. 2012. 3D binaural audio capture and reproduction using a miniature microphone array. In *Proceedings of the 15th International Conference on Digital Audio Effects (DAFx)*. 151–154.