Proceedings of the first research track of GameSoundCon: The Art, Technology and Business of Game Audio

September 27-28, 2016
Millennium Biltmore Hotel, Los Angeles, CA

CONTENTS

Spatial Audio Modelling to Provide Artificially Intelligent Characters with Realistic Sound Perception

B. Cowan[1], B. Kapralos[1], and K. Collins[2]
[1]University of Ontario Institute of Technology. Oshawa, Ontario, Canada.
[2]University of Waterloo. Waterloo, Ontario, Canada.

{brent.cowan, bill.kapralos}@uoit.ca collinsk@uwaterloo.ca

Abstract—Despite the importance of sound in the real-world, particularly when visual cues are not available, the non-player character's (NPC's) sense of hearing is typically ignored completely, and if present, it is simply distance based without taking the environment and effects such as diffraction into account. Here we present a method that employs acoustical occlusion/diffraction modeling methods to better simulate the hearing of NPCs in virtual environments including games. We provide NPCs the ability to better "perceive" sounds and therefore allow them to behave in a more natural, and realistic manner.

Keywords— Acoustical occlusion; acoustical diffraction; spatial sound; artificial intelligence; video games; virtual environment.

I. INTRODUCTION

The game industry acknowledges that effective artificial intelligence (AI) is a "necessary ingredient to make games more entertaining and challenging" [1]. Yet some video game players (gamers) are generally dissatisfied with the quality of game AI, and prefer human-controlled opponents [2]. While it is generally true that visuals/graphics sells games, poor AI is something gamers routinely complain about, and this in turn can negatively affect sales. The most common complaint about game AI is that it fails to behave in a believable manner. The lack of realism is in part because NPCs do not perceive the world in a realistic manner. In order for NPCs to behave realistically, their knowledge should be limited to that which they could perceive by way of their five senses (hearing, sight, touch, smell, and taste). This includes prior knowledge, current sensory input, and information received through some form of communication (e.g., from other NPCs, video surveillance, and alarm ringing, amongst others). However, typically only the visual sense is simulated, and this simulation is typically limited to checking for line-of-sight between the NPC and the player's avatar using simple ray-casting approaches. The NPC's sense of hearing is often ignored completely, or simply distance-based without accounting for the environment. To simulate the NPC's sense of sight, it is important to test for visual occlusion (blocking caused by objects in the environment). Similarly, acoustical occlusion/diffraction effects must also be approximated in order to simulate the NPC's sense of hearing.

Inspired by our previous work that saw the development of a graphics processing unit- (GPU-) based acoustical occlusion method used to approximate occlusion/diffraction effects for dynamic and interactive virtual environments and games [3], here we apply this method to NPCs to provide them with the ability to "perceive" sounds and therefore behave in a more natural, and realistic manner.

II. BACKGROUND

Our previous work saw the rendering of spatialized sound for a human listener based on a two dimensional virtual environment (or 3D environments that can be represented by a 2D mapping) [3]. The method begins by computing two 2D *collision maps* (e.g., two images/matrices). Each location in the first map (*resistance map*) stores a value representing the resistance to sound propagation at a location in the environment. In other words, it represents the ability of sound to pass through the space (an "occlusion value"), whereby solid objects appear white while empty space appears black, and everything in between grayscale. For a static environment, the collision map remains static. The second collision map (*distance map*) represents a distance field whereby each pixel represents the distance to the nearest collidable surface. This map is used by the GPU algorithm to allow for efficient ray-marching (parsing) of the environment. Based on these collision maps and the position of the player (the player's avatar), our GPU-based method generates four "information" output maps at interactive rates. These four maps provide a lookup table for i) the distance travelled (the distance from the player based on the shortest path), ii) the direction from which sound would reach the player, iii) the amount of occlusion as perceived by the player, and iv) the perceived direction to the player (sound direction) from every point in the environment. The data returned by the information output maps is completely independent from the location of any sound source; any location in the map may contain a sound source. The information output map lookup tables can then be used to acquire data regarding any sound source in the environment, thereby allowing for hundreds of dynamic sound sources. The distance travelled by each sound is based on the route taken by the sound in order to reach the player's avatar thus allowing for more accurate sound attenuation modeling.

The primary goal of the method is to use the output maps as lookup tables. Data that is retrieved from the information map lookup tables can be used to place a virtual sound source in an audio application programming interface (API) such as the Fmod audio engine [4] based on the relative distance, perceived direction, and the occlusion factor of the sound. Acoustical occlusion/diffraction effects are approximated by comparing the straight line distance between the sound source and the

listener with the path distance (the distance traveled by the sound). This simple comparison reveals whether the path taken by the sound emitted by the sound source will reach the listener directly or indirectly. The straight line distance and the distance of the path traveled are equal in the absence of occluding objects directly between the sound source and the listener. It is also possible to calculate the perceived direction and distance to the player (player direction) from every point in the environment. In this way, the method is able to simulate human-like perception of sounds originating at the player's location for any number of NPCs. By following the perceived direction of sounds made by the player (originating at the player's location) NPCs are effectively performing pathfinding (locating the shortest route between two points), assuming there are no obstacles to travel that are not obstacles to sound. The method is essentially a GPU-based pathfinding technique capable of calculating the path (taken by sound) from every point in the environment to the player, and from the player to every point in the environment simultaneously and in real-time [3].

III. Acoustical Modeling of NPC Perception

A. Overview

The 2D occlusion/diffraction algorithm described above provides four key pieces of information that can be used to simulate an NPC's ability to hear and locate sound sources: i) the shortest path that sound could travel in order to reach each NPC (listener) (D_p), ii) the actual straight-line distance (D_a) between sound source and each listener, iii) the acoustical occlusion value (described below), and iv) the direction that sound travels to reach the listener. The acoustical occlusion (O) is represented as a value between 0 and 1 denoting the approximate percentage of sound (perceptual) that is able to reach the listener. The direction vector is based on the "last leg of the journey" and is aliased based on the listener's immediate surroundings.

Aside from these four factors, our method also accounts for other environmental factors including the acoustics of the room and reverberation, in-game ambient sound, and occluding objects in the path between the sound source and the listener. Sounds are altered by their environment and these alterations can provide information regarding the sound's origin. A listener in a familiar environment may know which room a sound originated from based on the acoustics of the room, even if the path between the listener and sound source is indirect. Reverberation can have a negative effect on the listener's ability to locate a sound source [5]. More specifically, reflections with a short delay and amplitudes similar to the direct signal interfere with our ability to accurately localize sound sources [6]. However, the ratio of direct to reverberant energy provides an important distance cue for sound sources that are greater than one meter from the listener [7]. In addition, reverberation may provide additional information about the environment where the sound originated, such as the size of the room and the reflectivity of the walls and other surfaces.

Having additional information about the environment such as the average reflectivity of the materials present or the "room size", allows the system to estimate how useful reverberation is for sound localization. Room size is a term used by sound APIs such as Fmod. It is a value that ranges between zero and one, and it is used to describe how open or enclosed an environment is. For example, a room size of 1 represents an open space, while a small room size represents a small enclosed space such as a hallway or a tunnel. Fmod uses a room size estimation to simulate dynamic reverberation. A GPU-based algorithm that estimates both reflectivity and room size in real-time for use with Fmod and other sound engines is described by [8]. Background noise and acoustical occlusion can also have an effect on the sound that the listener hears by reducing the listener's ability to detect a sound, in addition to interfering with their ability to accurately locate a sound source. All of the factors discussed above were taken into account when designing a system to approximate human hearing for artificial characters in a virtual environment.

B. Implementation Details

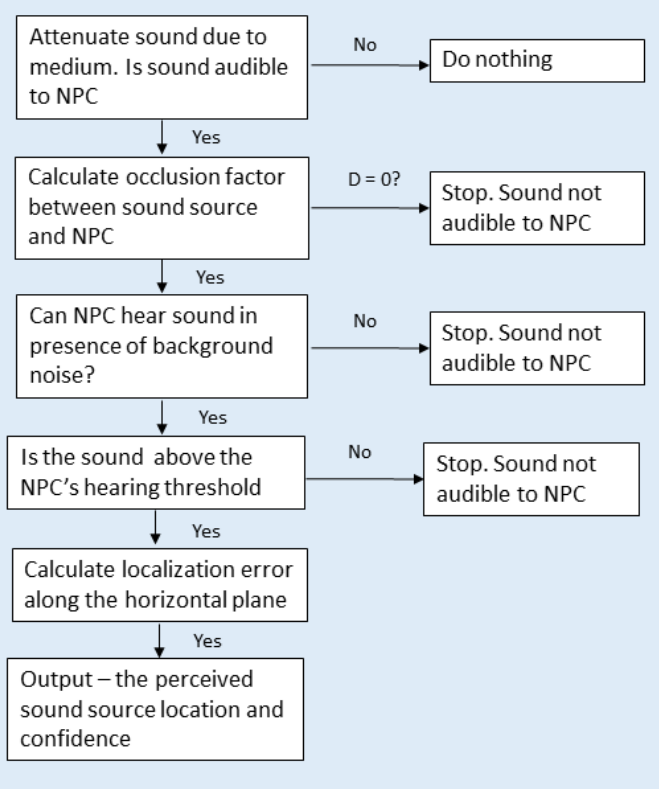


Fig. 1. Graphical summary of the NPC hearing method.

A graphical summary of the NPC hearing method is provided in Fig. 1. First, the amplitude of the sound is decreased to account for propagation of the sound through the air following the inverse square law. The amplitude is then scaled to account for the occlusion between the sound source and the listener (the NPC). The occlusion scaling factor is calculated using our 2D GPU-based occlusion/diffraction method described earlier. An occlusion value of 1 implies that the source is unoccluded, a value of 0 implies that no sound is able to reach the listener, and a value in between indicates that a portion of the sound is occluded only. The sound energy is further attenuated to account for both background noise (which can dull our sense of hearing making sounds appear less intense [9]). The next step is to estimate the localization error based on the intensity at the listener's position, the direction that the listener is facing, and

reverberation (reverberation can lead to an increased localization error due to the reflected sound waves carrying conflicting directional information [10]). Localization error is decreased as the sound intensity increases up to approximately 70 dB [11]. It has been suggested that sound source localization error along the horizontal plane is between 8 and 12 degrees under ideal conditions (young listeners seated in an anechoic chamber), with greatest accuracy when the sound source is directly in front of the listener [12].

NPCs can use the output from this algorithm to search for the source of the sound in a more realistic way. In a graph system, the environment is simplified for the NPCs by representing it as a series on interconnected nodes which can be searched efficiently using a path finding algorithm such as A*. The output from our method described above provides an area to be searched. Nodes located close to the center of the search area have a higher probability of containing the sound source (the player). Nodes outside of the search area would be given a probability of zero or a probability close to zero. The NPC could use this information to prioritize unsearched nodes based on the probability of the sound originating at that location, and their distance from the node. Nodes are considered searched if the NPC has an unobstructed line-of-sight with the node which results in the probability at that node being set to zero (there is zero chance that the player is present there). The NPC might return to a more relaxed patrolling state once all of the nodes have been searched.

IV. Summary and Future Work

Here we introduced a method that allows NPCs within a virtual (game) environment to perceive sounds in a manner similar to human hearing. In contrast to the majority of the current NPC technologies, our method allows NPCs to behave in a more natural and realistic manner. The accuracy of sound source estimation is affected by the game environment and is dependent on the shape of the occluding objects between the sound source and listener. Acoustical occlusion, diffraction, and attenuation are combined with environmental attributes such as openness (room size), reflectivity, and the background noise level to determine how the environment will influence the audibility of a sound. This result is compared with listener specific traits such as the direction the listener is facing and their overall hearing ability. This method operates in real-time (interactive) rates regardless of the number of NPCs that may be listening. Future work will incorporate the method into a game to provide NPCs with more realistic artificial intelligence and thus allow them to be behave more naturally. This can then be followed with a human participant-based study to gauge what effect this has on the user's experience with the game.

Acknowledgments

The financial support of the *Natural Sciences and Engineering Research Council of Canada* (NSERC), and the *Social Sciences and Humanities Research Council of Canada* (SSHRC) is gratefully acknowledged.

References

[1] Bowling, M., Fürnkranz, J., Graepel, T., & Musick, R. (2006). Machine learning and games. *Machine Learning*, 63(3), 211-215.

[2] Schaeffer, J. 2001. "A Gamut of Games." *AI Magazine*, Vol. 22 No. 3, pp. 29–46.

[3] Cowan, B., & Kapralos, B. Interactive Rate Acoustical Occlusion/Diffraction Modeling for 2D Virtual Environments & Games. In *Proceedings of the 6th International Conference on Information, Intelligence, Systems and Applications (IISA2015)*, Corfu, Greece, July 6-8, 2015.

[4] Fmod Audio Engine by Firelight Technologies. http://www.Fmod.org/.

[5] Zannini, C. M., Parisi, R., & Uncini, A. (2011, July). Binaural sound source localization in the presence of reverberation. In *Proceedings of the 17th International Conference on Digital Signal Processing (DSP)* (pp. 1-6). IEEE.

[6] Ribeiro, F., Ba, D., Zhang, C., & Florêncio, D. (2010, July). Turning enemies into friends: Using reflections to improve sound source localization. In *Proceedings of the Multimedia and Expo (ICME), 2010 IEEE International Conference on* (pp. 731-736). IEEE.

[7] Shinn-Cunningham, B. G. (2000, December). Distance cues for virtual auditory space. In *Proceedings of the IEEE-PCM* (Vol. 2000, pp. 227-230).

[8] Cowan, B., & Kapralos, B. A real-time, GPU-based method to approximate acoustical reverberation effects. *Journal of Graphics, GPU, and Game Tools*, 15(4):210-215, 2011.

[9] MacPherson, E. A., & Middlebrooks, J. C. (2000). Localization of brief sounds: effects of level and background noise. *Journal of the Acoustical Society of America, 108(4)*, 1834-1849.

[10] Giguere, C., & Abel, S. M.. (1993). Sound localization: effects of reverberation time, speaker array, stimulus frequency, and stimulus rise/decay. *Journal of the Acoustical Society of America, 94 (2)* , pp. 769-776.

[11] Davis, R. J., & Stephens, S. D. G. (1974). The effect of intensity on the localization of different acoustical stimuli in the vertical plane. *Journal of Sound and Vibration*, 35(2), 223-229.

[12] Carlile, S. (1996). Virtual Auditory Space: Generation and Applications. RG Landers.

3ME- A 3D Music Experience

A.F. Genovese, C. Craig Jr, and S. Calle
New York University, New York, United States.

jsc369@nyu.edu

Abstract—3ME is an interactive musical experience combining the potential of binaural recording techniques with the novelty of interactive 3D virtual reality environments. The user is placed in the center of live performance settings including vibrant orchestras, studio recordings, and improvisational ensembles. Full user immersion is achieved through the use of 3D audio recordings complemented by 360 video. Combining the audio and video elements in Unity, an Oculus Rift VR headset can precisely display the fully recorded visual sphere and track the listener's sound perspective within the scene. Differently to current standard implementations of 3D sound for VR, the methods explored in this project allow for the full capture and integration of the natural acoustic environment of a musical event or soundscape of interest. This project is the fruit of a collaboration of graduate and undergraduate students who participate in the NYU immersive audio interest group at NYU Steinhardt.

Keywords— Binaural; 3D; audio; VR; sound; Ambisonics.

V. INTRODUCTION

The use of 3D sound in Virtual Reality is relevant to a large range of applications such as video games, entertainment related content, live-event streaming, art installations, virtual music composition and collaborations, and educational environments. Current techniques rely on the use of Head Related Transfer Functions (HRTFs, see section 1.1) to simulate 3D binaural sound for an arbitrary number of sound-objects. Although the method of simulation of 3D sound is flexible and less constrained, it implies a complex sound production process and, in certain applications, misses out on the benefits of directly recording a representative sound field.

3M E was born as a follow-up project to the work described in [3] which aimed to capture 360 soundscapes of New York for multichannel or binaural reproduction. The integration of sound field recording techniques to Virtual Reality settings, opens the doors to new possibilities in terms of user immersion in sound-enhanced environments. The main advantages of the proposed methods involve the possibility of capturing the natural sound environment of a recorded event (for example the acoustics of a concert hall or a church) and a direct, easy and fast implementation of 3D sound for Virtual Reality that can significantly simplify the production stages of building a plausible binaural environment.

VI. BACKGROUND

The human brain is able to localize sound in space through the combination of binaural cues received at the two ears. Those cues are decoded by the brain to determine angle, elevation and distance of a sound source in space. The main cues are ITD (Inter-aural Time-arrival Difference), ILD (Interaural Level Difference) and spectrum. Direct 3D sound recording techniques aim to re-construct the binaural cues of the sound environment to the listener's ears. The recording techniques used for this project makes use of binaural dummy heads and sound field recorders that can capture full 360 soundscapes.
Various technologies were used to record, process, and reproduce binaural and ambisonic soundscapes. Professional binaural microphones such as the 3Dio Free Space Pro II and Omni binaural microphone were used to record performances from the perspective of human listeners. The Core Sound TetraMic recorded first order ambisonics, which allowed for the creation of more immersive environmentally focused recordings. Processing of the resultant signals was done using the spectral correction files and proprietary software included with the TetraMic; the VVMic. The software converts A format ambisonic signals captured by the TetraMic into B format, allowing for their eventual stitching together in any DAW (such as ProTools). The recorded audio was synced with video taken at the time of the recording using the Theta S 360 camera. Both the audio and the video were imported into Unity, where the Oculus SDK was used to enable the enjoyment of the recorded audio and video via Oculus' VR headset.

While binaural audio recorded through dummy heads is meant for headphone reproduction, sound field microphones may use Ambisonics techniques to deliver both binaural virtual surround-sound or loudspeaker multichannel reproduction. In contrast, popular simulation methods to achieve binaural 3D sound for headphones make use of HRTFs to create sound objects in VR. HRTFs are in fact pre-recorded FIR filter pairs (Left and Right ear channels) used to transfer the binaural cues of a particular location in space to any mono or stereo sound file, which will be then localized to the same spatial location. In addition, the room properties related to where the used HRTFs are recorded, such as room reverberation and absorption responses are also transferred to the sound (unless recorded in an anechoic space).

VII. METHODS

A. 3Dio Omni Pro

The dummy -head method captures spatial audio through electret capsules placed within plastic ear molds that recreate the binaural cues at the ears, providing an in head immersive experience when reproduced over headphones. In contrast to single-perspective traditional dummy heads, the 3Dio Omni Pro binaural microphone (Figure 1) consists of four ear pairs

pointing in four opposite directions (0, 90, 180 and 270) separated by a 90 angle. In Unity the sound from the 4 pairs is interpolated across the horizontal plane to form a navigable VR sound scene. The interpolation is done linearly according to where the user is looking in the 360-azimuth plane. Depending on this, the interpolation is done with the two stereo audio files taken from the quad recordings that are involved in that perceptual location. This means that if the user is looking at an angle of 45 in the azimuth plane, the signal from the dummy head looking at 0 and 90 will be combined at an intensity of 0.5 each, creating the illusion that the audio sources in the scene moves smoothly according to the picture. This technique achieves very good results in the azimuth plane, but as there is no possibility to do interpolation to achieve the same effect on elevation, the sound sources don't move when the user tilt the head up or down. This could be corrected with filtering to simulate the elevation of a source.

Figure 1. 3Dio Omni - 4 ear pairs point to the cardinal directions. The Theta S camera for 360 video recording is mounted on top

B. Ambisonics

Ambisonics uses spherical harmonics to represent the sound field. It can decode signals to a wide number of speakers and exists in a succession of formats; A, B, C, and D. A-format corresponds to the original audio picked up by the microphone. That audio can be transformed into B-format through the use of proprietary software or smart signal routing in a DAW. While C-format deals with transmission of Ambisonics signals, D-format deals with decoding and reproduction of the recorded sound. For our project, first order (Figure 3) Ambisonics recordings were obtained through the use of the Tetra Microphone by Core Audio. (Figure 2) The 4-cardioid microphone capsules capture audio in the X, Y, and Z planes. The output audio uses Ambisonics to represent sound in both the horizontal and vertical planes. Using different combinations of the capsules after B-format transformation [6], the sound is processed into a 16-channel virtual surround-sound setup (Figure 4).

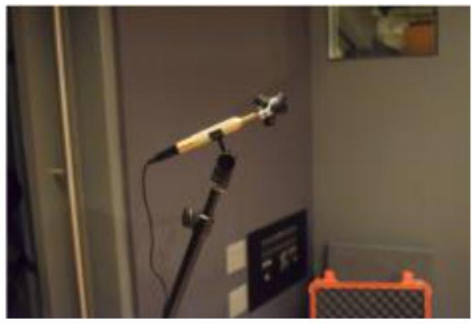
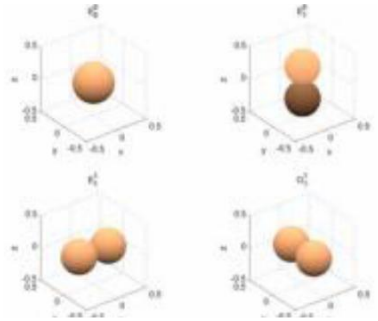


Figure 2. Tetramic by CoreAudio for Ambisonics recording



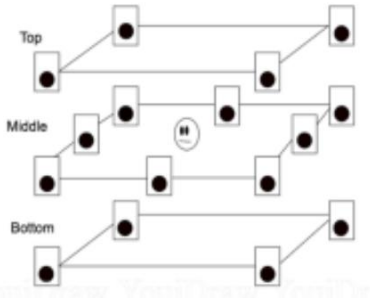


Figure 3. 0th and 1st order Ambisonics harmonics

Figure 4. 16-channel virtual surround-sound setup

VIII. VR IMPLEMENTATION

The 360 video recording is currently being done using the Ricoh Theta S 360 camera which records at a resolution of 1920 x 1080 pixels. The source material, is brought into Unity by creating an inverted sphere where the video is played as a movie texture. The Unity audio implementation differs according to the recording technique. For the 3Dio microphone the HRTFs are already printed in the recorded audio by the ear molds in the microphone. In this case, a linear interpolation is being done between the 4 stereo files in real time by taking into account the user's head rotation. For the Ambisonics the audio

is first transformed into a B-format file, which contains the W, X, Y and Z components. This is done by applying the following additions and subtractions to the four recordings extracted in the A-format:

$$X = LF - LB + RF - RB$$
$$Y = LF + LB - RF - RB$$
$$Z = LF - LB - RF + RB$$
$$W = LF + LB + RF + RB$$

MATLAB code is used to generate 16 Channels virtually placed into Unity according to the location of the recorded space, and rendered using virtual generic HRTFs (RealSpace3D audio plug-in). A mixing process can be done to the channels by separate if a different effect other than realistic sound image wants to be achieved. Inside unity we are using the GearVR and Oculus SDK 2 rotation information to render the audio in real time. This technique has the advantage of presenting height information, while the 3Dio interpolation is only done in the azimuth plane (height information will be further introduced using filtering). The final experience is finally uploaded into VR headsets accompanied by a pair of headphones.

IX. CONCLUSIONS AND FUTURE WORK

3ME is a project currently under progress that aims to explore and refine binaural recording and simulation techniques for virtual reality implementations. Currently the focus is on refining the capturing methods and establish a relationship between the type of sound setting, the application requirements and the optimal recording method to be used. This project illustrates the feasibility of successful VR implementation of binaural sound field recording methods that allow for easy and direct integration of the natural acoustics of a setting of interest into a virtual reality experience. Current issues include video resolution quality and the absence of height information for dummy head recordings. Another issue is the impossibility of mixing musical sound sources post-recording, meaning that great care must be placed in optimizing the levels of the instruments in relation to the microphone during the initial recording.

ACKNOWLEDGMENTS

This project was developed and implemented by the NYU Immersive Audio Interest Group. List of members: Agnieszka Roginska, Andrea Genovese, Christal Jerez, Jason Rostkowski, Charles Craig Jr., Chi So, Juan Simon Calle, Kiran Kumar, Jono Califa, Mert Cetinkaya, Zhe Sheng, Arun Pandian, Aggie H. Tai, Spencer Oliphant, Christy Welch, Christopher Miller, Jordan Juras.

REFERENCES

[1] Aswathanarayana, S., and Roginska, A. 2014. I hear ban-galore3d: Capture and reproduction of urban sounds of Bangalore using an ambisonic microphone. In 20th International Con- ferrous on Auditory Display (ICAD2014).

[2] Begault, D. R., and Trejo, L. J. 2000. 3-d sound for virtual reality and multimedia.

[3] Boren, B., Andreopoulou, A., M usick, M ., M ohanraj, H., and Roginska, A. 2013. I hear ny3d: Ambisonic capture and reproduction of an urban sound environment. In *Audio Engineering Society Convention 135*, Audio Engineering Society

[4] Follett, P. B. 1974. Ambisonic reproduction of directionality in surround-sound systems. *Nature 252*, 5484, 534–538.

[5] Laitinen, M . V., Kuech, F., & Pulkki, V. (2011, M ay). Using spaced microphones with directional audio coding. In *Audio Engineering Society Convention 130*. Audio Engineering Society .

[6] Malham, D. G., and Myatt, A. 1995. 3-d sound spatialization using ambisonic techniques. *Computer music journal 19*, 4, 58– 70.

Audiovisual Alchemy

Paul Hembree, PhD
1427 Saratoga Rd Apt 8
Ballston Spa, NY, 12020
+1 541 683-1779
paul.j.hembree@outlook.com

ABSTRACT

Audiovisual Alchemy is a novel method of procedurally generating music and abstract animation with potential applications for virtual and augmented reality games. It was initially developed as an interactive, multi-modal environment for improvising audiovisual experiences within the context of electroacoustic concert music. *Audiovisual Alchemy* combines techniques and theories from procedural pattern generation, spatial audio, and musical cognitive science to viscerally induce the experience of the technological sublime in users and audience members.

Concerning specific techniques, cellular automata are used in *Audiovisual Alchemy* to activate an array of sound- and light-producing modules within an explorable, virtual 3D space, rendered within a game engine. In essence, the cellular automata "play" the array of modules as if the array was an instrument. The spatial arrangement of these modules effects the potential musical pitches available to the listener at any moment, through the use of spatial audio techniques. Utilizing musical cognitive science, various arrangements of modules can be generated, enabling anything from consonant, traditional harmonies, to dissonant or even microtonal harmonies. The harmonic and melodic content produced by *Audiovisual Alchemy* is an emergent feature the activity of the cellular automata, the spatial arrangement of modules, and the guidance of the user.

CCS Concepts

Applied computing → Sound and music computing
Applied computing → Media arts
Applied computing Law, social and behavioral sciences → Psychology
Computing methodologies → Mixed / augmented reality
Computing methodologies → Virtual reality

Keywords

Audiovisual Software Instrument; Spatial Models of Musical Pitch Perception; Cellular Automata; Procedural Generation; Music Psychology; Music Theory

1. INTRODUCTION

Various researchers working on spatial models of musical pitch perception have provided computer musicians and software developers with a wide variety of potential multi-dimensional structures that could be useful for novel software instrument designs. These designs might specify instruments in traditional gaming or virtual reality platforms, taking advantage of the flexibility of modern game engines to distribute interactive, sound- and light-emitting components in virtual three-dimensional space. Such components might be equivalent to the velocity-sensitive keys on a traditional synthesizer, or they could capture more dimensions from user interactivity. Furthermore, with the incorporation of exaggerated or even surreal spatial audio processing, combined with procedural pattern generation techniques such as cellular automata, these instruments could be programmed to generate musical content automatically.

2. SPATIAL METAPHORS OF PITCH AND HARMONY PERCEPTION

Both cognitive scientists interested in music perception, as well as composers and music theorists interested in psychology have proposed a variety of spatial models of the perception of pitch and harmony within Western common practice music and its descendants, such as the tonal style pervasive in popular, film, and game music today. Composer and theorist Fred Lerdahl defines pitch-space theories as spatial models that are "intended to capture the sense of proximity and distance among pitch configurations that listeners bring to bear when hearing tonal pieces" [10, pg. 315]. Whether or not this sense of distance among pitch configurations is innate, or learned through exposure to Western common practice music, is not within the scope of this paper.

Several of these theorists, including David Lewin, Richard Cohn and Dimitri Tymoczko work within a framework first established by mathematician Leonhard Euler (1707 – 1783), and later developed by music theorist Hugo Riemann (1849 – 1919) [18, pg. 10-11]. This branch is sometimes referred to as Neo-Riemannian music theory, and involves work around a pitch lattice built on triads called the *Tonnetz*.

Coming from psychology and cognitive science, prominent pitch-space theorists include Carol Krumhansl, Diana Deutsch, and Roger Shepard. Most of their models were discovered through experimentation, though what they found has often corresponded with structures already known to music theorists.

Most pitch-space theories attempt to correlate the sensation of relatedness between individual pitches, or groups of pitches, with spatial distance, where relatedness is typically inversely proportional to spatial distance. For instance, Krumhansl developed three spatial models for cognitive distances, based on multidimensional scaling of empirical data, between individual pitches, between chords within one diatonic region (within one key), and between different diatonic regions (between keys) [7, pg. 28]. Her space of diatonic regions can be a useful heuristic for predicting the shock experienced when abruptly modulating between keys.[1]

1 Although I have engineered this paper to be useful to people of many backgrounds, readers may find it helpful to

By combining pitch space theories with a platform for first-person, virtual three-dimensional exploration and spatial audio processing, such as can be found in modern game engines, a software developer can create audible geometrical structures. In *Audiovisual Alchemy*, these structures are composed of light- and sound-emitting modules, whose location in space determines the pitch emitted, and perhaps also the color. Movement through, or manipulation of these modules, can be used to create dramatic and immersive audiovisual experiences, complete with harmonic progressions emergent from features of the spatial constructions.

Though it might be tempting to seek out a singular, universally potent spatial model of pitch, it is perhaps better to explore a variety of spaces, each with its own harmonic nuances. Many of these spaces can be procedurally generated using intertwined helical chains of pitches – a coincidental but beautifully poetic similarity to the structure of DNA.

2.1 Shepard's Double Helix

One particular helical three-dimensional pitch structure, developed by cognitive scientist Roger Shepard entails forming a double helix by wrapping the two complementary whole tone scales as strands around a cylinder. Shepard arrived at helical representations of pitch as a way to spatially model higher dimensional musical relationships, within the single physical dimension of air pressure fluctuation frequency (perceived as pitched sound). He summarizes his efforts as:

> Rectilinear scales of pitch can account for the similarity of tones close together in frequency but not for the heightened relations at special intervals, such as the octave or perfect fifth, that arise when the tones are interpreted musically. Increasingly adequate accounts of musical pitch are provided by increasingly generalized, geometrically regular helical structures [15, pg. 305].

To create these structures, we wrap the single dimension of pitch into one or more helixes, essentially taking that single dimension and winding it up and around a cylinder. We then regularly sample points along those helixes in order to create spatial nodes for each musical pitch (and later, for each pitch-light module within *Audiovisual Alchemy*).

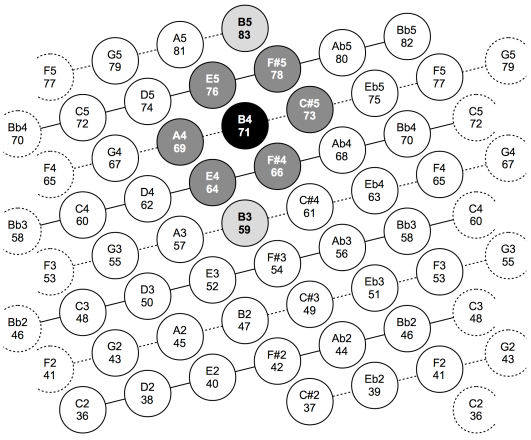


Figure 1: Two-dimensional representation of Shepard's double helix.

Shepard's double helix is depicted, sliced, unwrapped and flattened onto a two-dimensional surface, in Figure 1. The whole tone strand starting on C# starts rotated seven-twelfths further around around the cycle (and one semitone higher) than the C whole tone strand. Each pitch is marked with its note name and a MIDI note number, where middle C, C4, is 60. Dotted pitches show neighborly relationships that wrap around the cylinder; for instance, the right neighbor of F3 is G3, which is found on the other side of the cylinder. Every pitch is drawn vertically to scale, corresponding with its increasing logarithmic frequency.

The two helical strands, corresponding with the two complementary whole tone scales, are each marked with either a solid or a dotted line. The whole tone scale starting on C2 is solid, while the whole tone scale starting on C#2 is dotted.

To illustrate relationships between adjacent nodes, the nearest neighbors of the black pitch, B4, are marked in dark grey. In this neighborhood includes whole-steps (major seconds) connected horizontally by the helical strand, while diagonally up and to the right are fifths, and diagonally down and to the right are fourths. Octaves, in light grey, are not included in the local neighborhood.

2.2 A Vertically Compressed Double Helix

Spatial proximity in Shepard's original model does not correlate as well with traditional consonance (this was not his intention). In the original double helix, major seconds and minor sevenths, both mild dissonances in tonal music, have a spatial interval roughly equivalent to the perfect consonances of fourths and fifths. To remedy this, I found that compressing the helix vertically brings octaves into closer proximity than major seconds. All adjacencies within this model entail intervals of traditional perfect consonances: octaves, fifths and fourths.

Thus, in the compressed double helix, spatial proximity is roughly correlated with consonance. In both the original and compressed double helixes, increasing distance around the cylinder horizontally is correlated with traditional dissonance: the most remote pitches, on the opposite side of of the cylinder from each other, form intervals of minor seconds, major sevenths, and tritones. Vertically, increasing distance increases pitch height, represented by changes in octave.

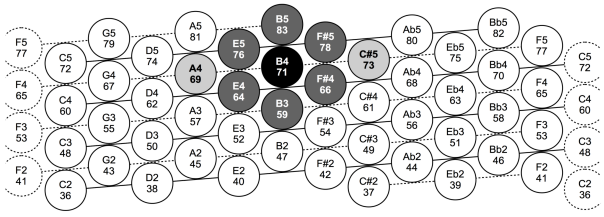


Figure 2: Vertically compressed double helix.

It is also more obvious in the vertically compressed helix that columns, which are all composed of a single pitch class, are separated by fifths or fourths from their neighboring columns. Thus, proceeding horizontally around the model entails moving

familiarize themselves with up to intermediate level music theory in order to maximize the utility of this paper.

around the traditional circle of fifths. When viewed from above, the columns in Figure 2 clearly form a circle of fifths, as depicted in Figure 3, which is also colored. I choose to divide the color wheel into twelve equal parts, assigning pitches to those colors according to each pitch's location in the circle of fifths. For the purposes of this paper I start with C as red, and move toward green as I move up the circle of fifths.

Because actual pitch-color synesthetia is both rare and highly subjective [3], I don't believe that there are any intrinsic perceptual connections between any particular pitch and any particular color. However, assigning color based off of the circle of fifths puts harmonically related pitches next to each other on the color wheel, as if to blend, and harmonically unrelated pitches across from each other, creating complementary color combinations that clash. It adds another orienting property to our figures.

One caveat when using the compressed double helix is that thirds and sixths, the imperfect consonances of tonal music, are represented by spatial distances greater than major seconds and minor sevenths, both mild dissonances (the same is true of the original double helix). By spatially reinforcing these earthy dissonances at the expense of the sweet imperfect consonances, using the compressed double helix for musical purposes tends to create harmonies reminiscent of modal jazz of the 1960s, or of the quartal and quintal harmony found in early twentieth century classical compositions.

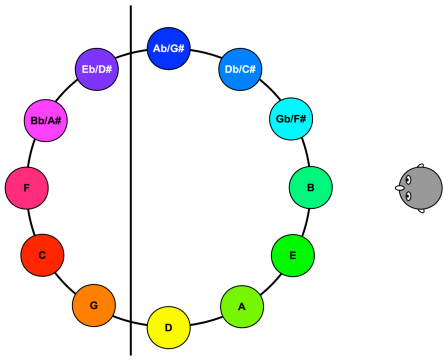


Figure 3: Colored circle of fifths, with hypothetical observer to the right.

Both helixes may be thought of as ways of spatially reconciling three factors: the perception of difference between adjacent frequencies; the perception of difference between octaves; and harmonicity [19, pg. 73], the perception of the relative consonance of pitch intervals.

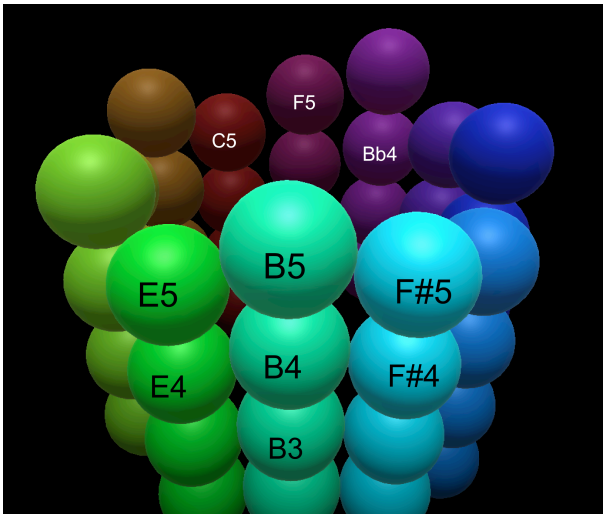


Figure 4: Compressed double helix rendered in 3D.

Figure 4 shows a three-dimensional representation of the structure depicted in Figure 2. Black pitch names are provided for several modules (rendered as spheres) in the neighborhood of B4, while white pitch names are for a selection of modules that are very harmonically and spatially distant from B4. Color is used as in Figure 3. Thus, nearby colors on the color wheel are harmonically proximate on the circle of fifths.

2.3 Spatial Audio as Articulative Mechanism

In order to experience the connection of spatial relationships with harmonic configurations, exaggerated spatial audio processing is used in *Audiovisual Alchemy* to attenuate and filter the sound of distant pitch-light modules. In my earlier software instrument *Apocryphal Chrysopoeia,* I used custom spatial audio programming written in MAX/MSP/Jitter 6. *Audiovisual Alchemy* now resides in the Unity 5 game engine.

In the domain of the eye, a fog effect is applied to attenuate the visual presence of those same distant modules. By doing so, it allows the observer to isolate and listen to the harmony on any particular side of the helix, while having the harmony on the opposite side of the helix less perceptually present, both visibly and audibly. This is depicted in Figure 3 with a line dividing the circle of fifths apart; the seven closest pitches to the observer form a diatonic collection, while the five furthest pitches form the complement to that diatonic collection, and are notes that would be outside of the key entailed by that collection.

Whatever modules are nearest the observer strongly influence the pitches available to the listener in what might be best described as a harmonic field. Were we to attenuate the amplitude of the distant modules in Figure 3 and 4, we might experience a harmonic field similar to that notated in Figure 5. Pitches are rendered darker when they would be louder, due to spatial audio processing. Note that B5 is the loudest, and darkest, while B2 is the quietest and lightest.

This is a good visual representation of a diatonic harmonic field typically encountered using the standard circle of fifths allocation of pitches in *Audiovisual Alchemy*. With the structure rotated to this side, the pitches conform to an A major scale, although it would likely be heard as a B dorian scale due to the prominent position (both in loudness and number of octaves represented) of the pitch class B, which would probably be heard as a local tonal center or tonic. I am intentionally not using the term dorian "mode" because "mode" implies a specific repertoire of melodic behaviors in addition to a repertoire of pitches.

If the user rotates the structure, the harmonic field modulates as new pitch classes are introduced at the perceptual periphery. If we were to rotate the entire structure, we would hear pitch classes added and removed very smoothly and subtly, as they become audible and fade away. This would be experienced as roughly equivalent to modulation of key within traditional tonal music.

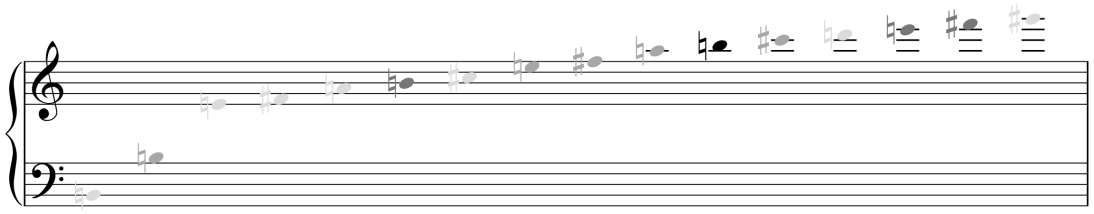


Figure 5: Harmonic field implied by Figure 3 and 4 (exact pitches)

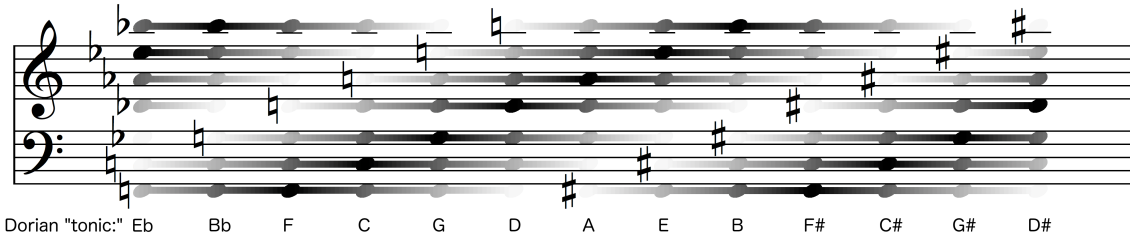


Figure 6: Successive harmonic fields created by rotation of the structure in Figures 3 and 4 (pitch class only)

Starting from the dorian collection centered on Eb, and ending on its enharmonic equivalent of D#, Figure 6 demonstrates how we might experience this smooth modulation. Again, the darker the notehead, the louder the pitch. Figure 6 utilizes spacing by fifths to show the orderly ascension of pitch class by half steps around the circle of fifths. However, this figure treats all octaves of a pitch class as equivalent; it simply shows which pitch classes would be present on any particular side of the structure.

2.4 Alternate Helical Arrangements of Pitch

In addition to the original and compressed double helixes, other helical models of pitch can be used. If the objective is not to model aspects of the perception of tonal music, but instead to creatively explore the possible relationships between space and harmony, no pitch space with a rigorous construction would be any more or less appropriate for than any other.

Every pitch space has its own truth that can be explored. Each pitch space provides affordances for particular harmonies, depending on the proximity of specific pitch modules to each other, in a way similar to the affordances provided by alternate tunings on guitar. For instance, the popular drop-D alternate tuning allows players strum an open D major chord, utilizing the lowest string as the root of the chord, a string which would otherwise be muted, providing a richer harmony. Similarly, alternative helical models of pitch bring different pitches into closer relationships with each other, allowing the user to tune the the particular flavor of harmony she wishes to encounter more frequently.

Using helixes as a foundation for pitch spaces is attractive because helixes lend themselves to procedural distribution. Unlike Shepard's helixes, many alternative helixes are non-isomorphic; that is, the same harmonic configuration may have different shapes depending upon the key.

Because the double helixes based on Shepard's work connect spatial proximity with traditional consonance, an interesting alternate model might do the opposite, connecting dissonance with spatial proximity. Wrapping the chromatic scale into a single helix, with six pitches per wrap, puts semitones on the horizontal axis, and tritones on the vertical axis, both dissonant intervals in traditional tonal music. This pitch-space is depicted in Figure 7.

pitch classes are still colored here according to their place in the circle of fifths, and their distribution in Figure 7 allows us to visually perceive the clash of the dissonant semitones and tritones as clashes of color. Connectivity around the helix is depicted with lightly colored pitches. Like the original helixes, this space is isomorphic.

Other alternate helical models include triple helixes based on wrapping diminished seventh chords into strands, and then interleaving them in different ways. With enharmonic equivalence, there are only three diminished seventh chords available within the equal-tempered system. Therefore, by transposing each successive helix up by a semitone, all twelve pitches are distributed with a triple helix of diminished seventh chords.

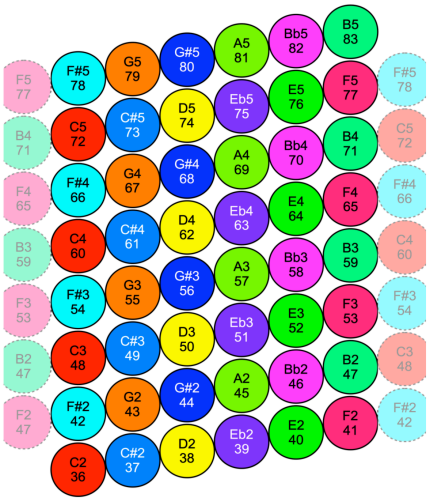


Figure 7: Dissonant Single Helix

In order to acquire more horizontal surface area, these helixes can be interleaved at various rotations for a variety of effects. One particularly attractive interleaving of diminished seventh chords is pictured in Figure 8, which creates an interesting hybrid of dissonance and consonance in connection with proximity. Some adjacencies entail minor seconds, while others entail fifths and fourths. This means that the structure is non-isomorphic, although it contains symmetries. The C# (heavy dotted line) and D (lightly dotted line) helical strands are rotated seven- and fourteen-twelfths of the way around the cylinder, with respect to the C strand (shown with a solid line). Again, the coloring is based on the circle of fifths, which shows us when the adjacent cells would blend through consonance, and when they would clash through dissonance.

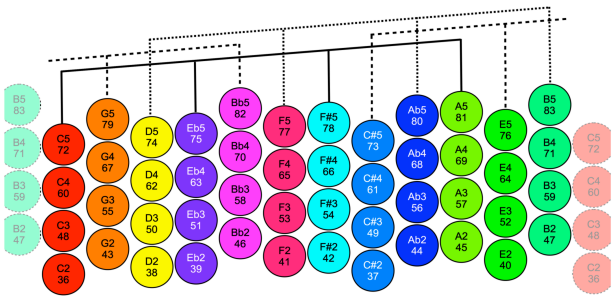


Figure 8: Triple Helix with Mixed Intervals

2.5 Further Methods of Articulation

I have demonstrated spatial structures of pitch-light modules informed by and expanding upon psychological models of the perception of musical pitch, and I have demonstrated that spatial audio can help articulate these structures. There remains a question, however, of how these audible structures are "played," beyond simply highlighting portions of them with spatial audio – do their constituent parts constantly emit sound, droning continuously, or are they articulated in time?

These spatial structures have the potential to serve as virtual instruments in augmented or virtual reality environments, where the articulation of pitches in time would be up to the user. Learning to play the isomorphic helical structures would have an advantage over learning to play, for instance, the piano, because all intervallic combinations would be identical no matter the key. This is not the case with piano, on which the shape of any chord changes depending on the key. The guitar is only partially isomorphic, with one major caveat.[2]

The principal problems with this instrumental model involve how to rotate the structure, and how to articulate pitches, using existing human interface devices such as the LeapMotion. Acoustic instrumentalists' natural desire would be to use individual fingers to articulate different pitches, but it even with LeapMotion's advances heralded by Orion, the fluent, tactile control over the device might not yet be possible – but it is coming soon.

Alternatively, *Audiovisual Alchemy* might be used purely for audio visualization. This could be approached in two ways, drawing directly upon a digital audio stream (such as linear PCM audio), or upon symbolic representations of music (such as MIDI). One could use real-time pitch detection, or more generally, real-time spectral analysis of recorded music to excite the states of pitch-light modules. Because the harmonic decisions would have already been made by preexisting musicians, the pitch-spaces would simply be used to spatially depict the pitch configurations within the music. Thus, spatial audio would not be strictly necessary. MIDI-driven articulation of the structure would be similar, with exactly known pitches, but with perhaps less appealing synthetic timbres compared to pre-recorded music.

I have principally used these pitch-space structures not for either manually controlled virtual instruments, nor for music visualization, but rather for procedural generation of music, with high-level control provided by the user. The user acts as an improvisor, an arbiter of dramatic, global musical decisions, but not local details. Because of their lattice-based construction, these pitch-spaces lend themselves to articulation with lattice-based generative processes, such as cellular automata.

3. CELLULAR AUTOMATA

Cellular automata (abbreviated hereafter as CA) are discrete systems whose global behavior is the emergent result of local interactions [17, pg. 5]. Important theorists of CA include John von Neumann (1903 – 1957) and Stanislaw Ulam (1909 – 1984) in the 1950's, John Conway in the 1970's, and Stephen Wolfram in his *A New Kind of Science* (2002) [21, pg. xi]. For a detailed introduction to CA, I would recommend particularly Joel Schiff's *Cellular Automata* (2008) [13].

CA typically operate on a uniform lattice of cells, in some number of dimensions; one- and two-dimensional CA have been the most studied. While higher dimensional CA are certainly possible, they can be computationally more expensive.

2 The standard tuning of the guitar involves one discrepancy in what would otherwise be a quasi-isomorphic space: the interval from the G to the B string is a major third, in a space of what would otherwise be all fourths. Furthermore, spatial distances corresponding with musical intervals shrink as a guitarist ascends the neck of the instrument, although their relative proportions stay roughly the same.

Each cell within a CA has a number of states; in the simplest case, just two states can be used, represented in binary. A variety of metaphors for these states have been proposed over the years, including active and inactive, on or off, living or dead. Furthermore, CA are usually modeled in discrete time steps, with updating occurring simultaneously across all cells (synchronous updating). Asynchronous updating is certainly possible, and may be more applicable to modeling chemical reactions [13, pg. 118-120], but the predictability of the system partially breaks down in such cases.

Each cell calculates its next state based on a transition function or ruleset, usually involving its current state, and the current states of one or more neighbors.

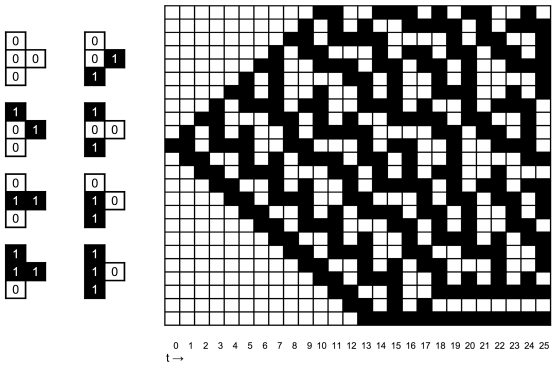


Figure 9: Wolfram's Rule 30, with 25 generations.

An example of a transition function can be found in Wolfram's one-dimensional Rule 30, in Figure 9. In each update cycle, every cell looks at its two neighbors' current states, above and below, and its own current state, and calculates its next state based on a set of eight rules, one for each possible combination of neighborhood states. Starting from a single active cell, with a state of 1, a chaotic but principled pattern is generated in just 25 generations.

CA can also use continuous states, represented with decimal values. In Figure 10, the transition function takes the average value of the three neighboring cells, adds 0.275 to that number, then takes the fractional part of the result. The added number, 0.275 in this case, can be thought of as an indicator of the reactivity of the system.

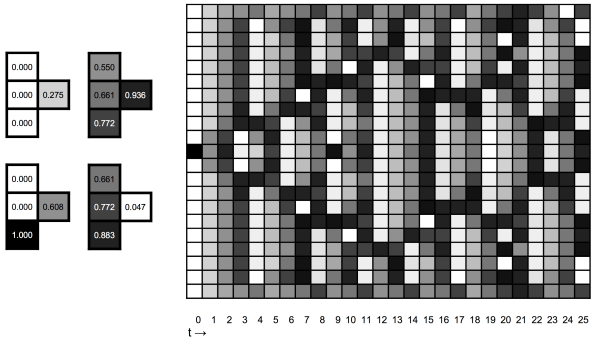


Figure 10: Continuous CA, with 25 generations.

The area outside the bounds of the simulation in this case count as an immutable value of zero. The results are similarly complex to Rule 30, and one can imagine using lower reactivity and higher temporal and state resolutions to obtain even smoother results. Using continuous cellular states can be beneficial for musical purposes, as we will see shortly.

CA are useful as engines for procedural generation because they can create extremely complex and "nature like" global features by iterating simple, local interactions [8][13]. CA have already been featured in games as terrain generators in *Galak Z* (2015)[1], or as fluid simulators in *Dwarf Fortress* [5], among others [6].

Cellular automata are attractive as tools for music generation because of the apparent propagation of cellular activity between adjacent cells. The cellular lattice might be mapped onto pitch, with the amplitude of each pitch activated by cell state. The cellular activity might spread across the lattice in a fashion similar to the spread of melodic lines driven by closest-motion voice-leading and contrapuntal rules.

CA differ, however, from the part-writing, linear model of musical behavior, because there is no concept of a musical line, or voice, in a CA: there are potentially as many voices as there are cells, assuming the lattice those cells occupy is mapped onto pitch. Therefore, CA tend to be more useful for generating masses of sound, in an aesthetic similar to that of the textural music of György Ligeti. As we will see, it was one of Ligeti's contemporaries that first applied CA to music.

3.1 Cellular Automata and Music

It is perhaps not surprising that the mathematically inclined architect-composer Iannis Xenakis was the first internationally prominent composer to use cellular automata. He used CA as a generative tool in his large orchestral works *Horos* (1986) and *Ata* (1987). In the Pendragon edition of his book *Formalized Music* (1992), he writes:

> Another approach to the mystery of sounds is the use of cellular automata... It is on the basis of sieves that cellular automata can be useful in harmonic progressions which create new and rich timbric fusions with orchestral instruments [23, pg. xii].

While Xenakis did not specify the exact CA or mapping method used in either *Ata* or *Horos*, researcher Makis Solomos found connections between Stephen Wolfram's article "Computer Software in Science and Mathematics," published in the August 1984 edition of Scientific American [20, pg. 188-203] with *Horos*, while researching the piece, discovering that that Xenakis used a pocket computer to calculate the successive steps in a one-dimensional, four-state CA detailed in Wolfram's article [16, pg. 9], mapping that CA onto music symbolically.

Xenakis took the non-zero cells, with values 1, 2 or 4, and mapped those cells onto note events for either woodwinds, brass or strings. Their position along the cellular lattice was mapped onto pitch, using a *sieve*, or dissonant scale, and the time-steps were mapped first onto even sixteenths, for one 4/4 measure, then onto asymmetrical rhythmic groupings for a longer passage. The results were typical of Xenakis' orchestral music: brutal, monolithic chords, articulated with mechanical precision,

expanding and contrasting erratically across the musical space.

This type of symbolic mapping of a one-dimensional CA onto a one-dimensional representation of musical pitch is a fairly normal strategy among composers interested in CA [11]. In a totally different, largely diatonic style, this method was even picked up outside the experimental computer music community by Wolfram Research in their *Wolfram Tones* (2005) project [22].

With one-dimensional CA mapped onto music, the pitch material can become dangerously static in its harmonic character, if the CA does not have complex emergent behavior. Furthermore, without any amplitude envelope management, the blocky character of binary CA creates hard distinctions between note-on and note-off events, mechanically in synchronization with the update cycle of the CA. This tends to create monotonous, motoric rhythms, that also tire the ear quickly. Xenakis himself could only stand mapping one measure of the CA simulation in straight sixteenths, before attempting to mix-up the rhythms in other ways, and Wolfram Research's system obscures the monotony of the CA update cycle by underpinning it with percussion loops that have no intrinsic relationship with the CA.

3.2 Combining Continuous Cellular Automata and Multidimensional Pitch Spaces

One solution to the harmonic stasis problem of simple CA mappings is to use the aforementioned multidimensional pitch spaces as the spawning ground for CA, and then to rotate or unpredictably explore these spaces to refresh the pitch material available at any one moment. This furthermore provides much more interesting formations for the eye, allowing for a complete audiovisual experience.

As a solution to the monotonous rhythm of CA update cycles, using continuous CA with a high temporal and amplitude resolution, and low reactivity, can help greatly, by smoothing the surface of successive update cycles. As an alternative to a high temporal resolution, gradually fading in and out note events can work as well, combined with randomly delaying the onsets of note events. A precedent for music generated with continuous cellular automata was set by MIT professor Chris Ariza in 2007 [2], and his idea of dynamically changing CA transition functions for dramatic effect during musical performance has been influential on my work.

Audiovisual Alchemy uses a simple continuous automaton, similar to that described above, with the decimal-value state mapped onto the linear amplitude of each pitch-module. Instead of working in one dimension, it instead works in three. Neighbor detection is done using a variable-sized sphere collider around each module, in Unity, which is an efficient manner of finding all adjacent modules within a radius. Counter-intuitively, a larger neighborhood involved in automata calculations actually reduces the density of the resulting activity.

The new state of each pitch-light module is calculated by taking the current average state of the neighborhood, including the state of the cell in question, multiplying it by a reactivity coefficient, adding a reactivity offset, then taking the fractional part. The result is inverted, within the range from 0 to 1, when mapping that value onto amplitude. Psuedo-code for this function is found below.

newState = fract((coef * curLocalAvgState) + offset)

newAmplitude = 1 - newState

The coefficient and offset each offer a slightly different way of managing the automata. Inverting the result creates distinct attacks, as the state abruptly flips from near 0 to near 1, that usually fade over time.

Currently, the structure of modules can be rotated left or right at a variety of speeds (Figure 11, bottom fader), and it can be viewed from above or below (left fader) from a specified distance (right fader). Though restrictive for now, these movements offer a controlled environment for observing the automata during development.

In previous generations of my software instruments, cross-modal mappings were found to be of vital importance for providing the audience with a richer experience than just a CA and movement through a spatial arrangement of pitches. In *Audiovisual Alchemy*, color is used to orient the user with regard to currently active pitches, and thus timbre, or tone-color, has no immediately obvious and available partner within the visual domain.

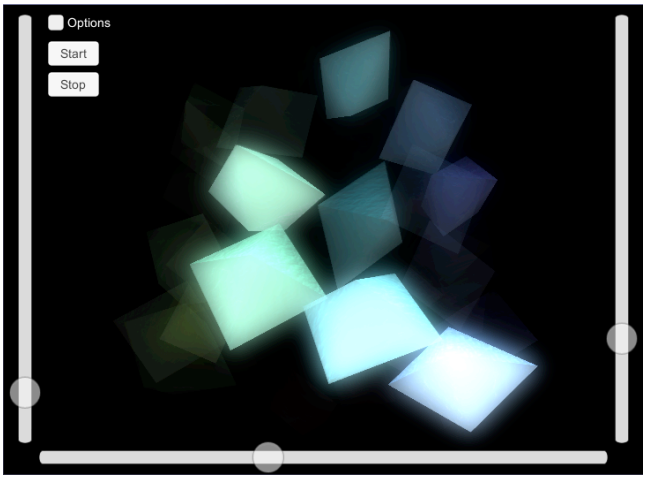


Figure 11: Screenshot of *Audiovisual Alchemy* alpha release.

Instead, the rotation of modules in three-dimensions determines the relative amplitudes of three simultaneously played waveforms, each based off of a particular family of instruments. For example, a module's rotation in the X dimension is mapped onto the amplitude of the brass waveform, the Y rotation to the amplitude of a string waveform, and the Z rotation to the amplitude of a woodwind waveform. Without a distinctive texture, the rotation of the spheres is not obvious, therefore, I typically use alternative geometric shapes for the modules, that more obviously betray their rotation. This provides a simple yet robust way of creating a visual and even quasi-physical connection with timbre. Further cross-modal mappings would be important for creating a full-fledged improvisation environment.

Audiovisual Alchemy currently offers some customizability under the hood, via an options menu (Figure 12). The default helical structure is the compressed double helix, but the exact structure used can be specified procedurally. The details of this process are beyond the scope of this paper. Furthermore, the neighborhood

size, reactivity coefficient, reactivity offset, rotation & tremolo speed, and update time interval can all be changed.

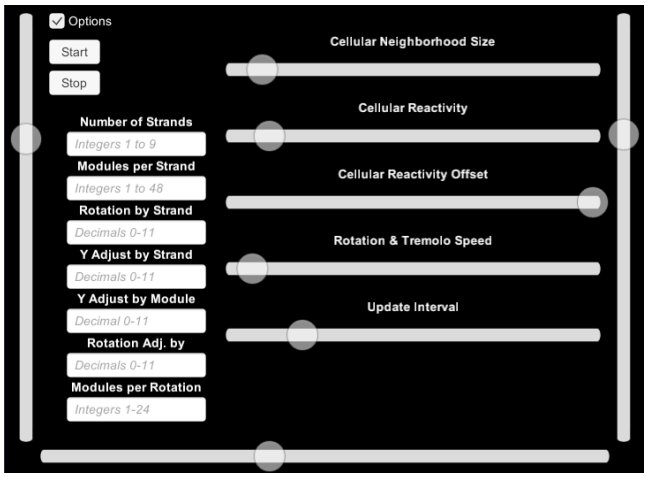


Figure 12: Alpha release options menu.

4. FUTURE PROSPECTS

Perhaps the most pressing concern for the further development of *Audiovisual Alchemy* is the programming of intuitive controls. There are numerous parameters that could effect both the exploration of the space and the behavior of the automata. Ideally, it would be best to strive for low-dimensional control over a high number of perceptual dimensions. Therefore, reducing the number of parameters available to the user through direct control by automating changes in some parameters might be the best solution. Simple cyclical automation, using low-frequency oscillators, or random automation, for instance using Brownian motion, might work very well.

Furthermore, the calculations involved in both constructing helical pitch-spaces and running cellular automata are opaque to the general user, so some options would have to be hidden from the user, or explained in a general way. "Reactivity coefficients and offsets," a "double helix of complementary whole tone scales," or a "semitone-tritone space" – all of this specialized language, though powerfully specific to the expert, would have to be reduced to descriptors such as pure "reactivity," or "consonant" and "dissonant structures."

Ideally, *Audiovisual Alchemy* would incorporate quasi-tactile control over the orientation of the structure of pitch-light modules, which is why virtual reality combined with hand tracking is the best build target. Virtual reality is also an ideal platform because developers are finding that VR tends to be suited to the manipulation of objects, while traditional gaming is more about the exploration of large spaces (larger than the user's living room). Jesse Schell, the developer of *I Expect You To Die* (2015) and professor at Carnegie Mellon University, recommends developing VR experiences that "make you feel like you are really in a place," allowing the user to "touch and manipulate objects," while avoiding "first person virtual motion" [12]. Many users of VR experience extreme nausea when the virtual camera moves, while the body stays still.

Creating a structure of modules that is human-sized or less, while allowing the VR user to rotate or resize this structure with her hands, would be ideal. Obviously, the spatial audio and fog effects would be exaggerated, with objects receding into the distance, visually and sonically, at just over an arm's length away. The end result would be a surreal, audiovisual sandbox, involving collaboration with the emergent behavior of cellular automata. The user might stand in a softly glowing fog, adjusting the reactivity of a pulsating helix of sound-emitting shapes, creating an arresting private experience, or even public spectacle.

5. REFERENCES

[1] Aikman, Zach. On Procedurally Generated Levels in Galak Z. In *Unite 2014* conference video documentation (Seattle, WA, August 26, 2014). Retrieved September 14, 2016: https://youtu.be/ySTpjT6JYFU.

[2] Ariza, Christopher. Automata Bending: Applications of Dynamic Mutation and Dynamic Rules in Modular One-Dimensional Cellular Automata. *Computer Music Journal,* 31, (1, 2007), 29-49.

[3] Brougher, K., et al. *Visual Music: Synaesthesia in Art and Music Since 1900*. Thames & Hudson, London, 2005.

[4] Chion, Michel, Claudia Gorbman, and Walter Murch. *Audio-vision: sound on screen*. Columbia University Press, New York, 1994.

[5] Harris, John. The Making of Dwarf Fortress. Interview with Tarn Adams. *Gamasutra* (February 27, 2008). Retrieved September 14, 2015: http://www.gamasutra.com/view/feature/3549/

[6] Harris, John. An Intro to Cellular Automation. *Gamasutra* (May 4, 2011). Retrieved September 14, 2015: http://www.gamasutra.com/view/feature/134736/

[7] Krumhansl, Carol L. Perceptual Structures for Tonal Music. *Music Perception,* 1 (1, 1983): 28-62.

[8] Kusch, Ingo and Mario Markus. Mollusc Shell Pigmentation: Cellular Automaton Simulations and Evidence for Undecidability. *Journal of Theoretical Biology,* 178 (3, 1996): 333 – 340.

[9] Lerdahl, Fred. *Tonal Pitch Space*. Oxford University Press, 2001.

[10] Lerdahl, Fred. Tonal Pitch Space. *Music Perception*, 5 (3, 1988), 315-349.

[11] Nierhaus, Gerhard. *Algorithmic Composition*. Springer, Dordrecht, 2009.

[12] Schell, Jesse. Six Lessons Learned From I Expect You To Die. *Gamasutra Blogs* (June 26, 2015). http://www.gamasutra.com/blogs/JesseSchell/20150626/247113/

[13] Schiff, Joel. *Cellular Automata*. John Wiley & Sons, Hoboken, NJ, 2008.

[14] Shanken, Edward A. Life as we know it and or life as it could be: epistemology and the ontology/ontogeny of artificial life. Leonardo, 31 (1998), 383-388.

[15] Shepard, Roger N. Geometrical Approximations to the Structure of Musical Pitch. *Psychological Review,* 89, (4, 1982), 305-333.

[16] Solomos, Makis. Cellular automata in Xenakis's music: Theory and Practice. In *Definitive Proceedings of the International Symposium Iannis Xenakis,* (Athens, May 2005-2006).

[19] Toffoli, Tommaso and Norman Margolus. *Cellular Automata Machines*. MIT Press, 1985.

[18] Tymoczko, Dimitri. The generalized Tonnetz. *Journal of Music Theory,* 56, (1, 2012), 1-52.

[19] Wishart, Trevor. *On Sonic Art*. Harwood Academic Publishers, Amsterdam. 1996.

[20] Wolfram, Stephen. Computer Software in Science and Mathematics. *Scientific American,* 251, (3, 1984): 188–92, 194, 196–200, 203.

[21] Wolfram, Stephen. *A New Kind of Science*. Wolfram Media, Champaign, IL, 2002.

[22] Wolfram, Stephen, et. al. Wolfram Tones: Web-based music software (Wolfram Research, Champaign, IL, 2005). Retrieved September 14, 2016: http://tones.wolfram.com/

[23] Xenakis, Iannis. *Formalized music; thought and mathematics in composition*. Indiana University Press, Bloomington, IN, 1992.

Auditory Immersion of 5.1 Virtualization within Gameplay

Sam Hughes
University of York
Heslington
York, United Kingdom
ssh508@york.ac.uk

Gavin Kearney
University of York
Heslington
York, United Kingdom
gavin.kearney@york.ac.uk

ABSTRACT

This paper addresses the level of auditory immersion felt by players engaged in a first person horror game with 5.1 binaural audio.. An experiment is presented where two sub-ject groups are asked to assess the immersion of 5.1 au-dio rendered over headphones using KEMAR head-related transfer functions (HRTFs) in comparison to a straight up two-channel downmix of the 5.1 stream. No statistically significant difference was found in the level of player im-mersion illustrating that 5.1 binaural rendering using non-individualised HRTFs does not enhance levels of auditory immersion in the game over headphone based stereo. The work represents a benchmark from which further potentially immersive technology such as head-tracking, personalized HRTFs and full sphere rendering can be evaluated during gameplay.

Keywords

Binaural Audio; Digital Games; Immersion; Localisation

1. INTRODUCTION

Recent developments in immersive cinematic surround sound have led to systems with increased channel counts that augment the traditional 5.1 surround setup, such as Dolby At-mos [5] and Auro 3-D [18] in an effort to deliver more im-mersive aural experiences to a large audience. For the in-dividual, such new audio formats could potentially enhance immersive games or virtual reality experiences by extend-ing the soundfield to three dimensions. For a long time 5.1 has been seen as the benchmark for immersive sound systems, however its capability for immersive audio rendering in the context of 'active gameplay' has yet to be assessed. Furthermore, a large majority of players will utilize headphones with 'virtual 5.1' as it is often impractical to have loudspeakers spread around a living room. There has been much previous research into levels of immersion [10], game engagement [2] and binaural audio [19] but there hasn't been any direct research into the effect that playing a video game with 5.1 binaural audio has on the player's level of immersion. This paper therefore poses the question 'Does virtualized 5.1 give the player an improved perception of auditory immersion over 2-channel stereo playback over headphones?' The question is put in the context of first person gaming and an experiment is presented to evaluate this under the condition of a horror game in which sound sources are localised around the player. This has been shown in prior studies by the author to have a direct impact on the amount of fear felt by the listener [7].

2. PRIOR WORK

Design of immersive binaural systems requires an understanding of the perceptual cues for sound source localisation. Any source at a given angle of incidence to the head will create subtle time and level difference cues at the ears and is subject to spectral shaping due to the pinnae [4, 13]. These cues are embedded in the Head Related Transfer Function (HRTF). For headphone reproduction of 3-D audio, filtering a source signal with a unique pair of HRTFs and presenting these filtered signals over headphones will ideally give the listener the impression that the source is located outside of the head and in the direction dictated by the filters. This process, known as binaural synthesis, has several drawbacks. First, HRTFs change from listener to listener and the capture of large datasets of HRTFs is expensive and time consuming. As yet there is no assured method for selecting 'near-match' HRTFs for an individual and generic HRTF sets, such as those measured with dummy head microphones, are known to produce sound localisation errors, including front-back reversals, and lack of externalization [1].

In terms of immersion, some research has been made into its relationship with game sound and a framework for the analysis of conceptual design of game audio in relation to immersion was defined by Huiberts [8]. This framework focuses on the structure and design process of game audio discussing the IEZA model in relation to immersion, yielding four domains; Interface, Effect, Zone and Affect. This

Figure 1: PS4 to DENON AV Receiver then into Traveler Mk 3 soundcard

study found there to be positive and negative influence of audio on immersion. The positive include audio enhancing sensory immersion, where either the presence (feeling as though in the game world)[15], or sensory gratification is enhanced. Huiberts also discusses how multiple speakers, surround headphones and other devices that emulate 3D sound heighten the feeling that the player is in the world of the game.

There is also substantial research into the impact of audio on player performance. Tan et al [17] found that people playing Legend of Zelda: Twilight Princess [12], performed worse when playing with both music and sound effects off. In an opposing study, Yamada et al.[20] found that people had faster lap times in Ridge Racer V [11] when racing with the music off. Another study also showed that in DOOM [9], male gamers scored almost twice as many points when playing with all of the sound on [16]. The effects can also be rather more complex. Cassidy and MacDonald [3] found that players driving with music and sound effects, when compared to just sound effects, drove much faster and were more aroused. However, they also made more mistakes.

3. THE EXPERIMENT

The aim of this experiment is to ascertain how different the immersive experience is between listening to downmixed stereo and virtualized 5.1 binaural. The hypothesis for this experiment is that there will be a significant difference in levels of immersion between players utilising the 5.1 binaural playback system whilst playing a horror game and players who are hearing the game audio in downmixed stereo. A further prediction is that binaural audio will actually increase the levels of immersion felt by the player.

The experiment was conducted in the Interactive Media Research Laboratory at the Department of Theatre, Film and Television at the University of York. This is an acoustically treated laboratory. The experiment utilized a PS4 system running the game Alien Isolation [14] through a standard HDMI connection to a 60 inch Panasonic LED TV. The Dolby digital audio output of the PS4 was routed via an optical cable into a Denon AV Reciever and converted to analog before being fed into a MOTU Traveller Mk3 soundcard to allow the virtualization and downmix process to affect the audio. The preamplifier gain on the soundcard was set to line level with the AV receiver at -20dBFS for a comfortable listening level with no distortion.

All participants were positioned with the centre of their eyes 46 inches away from the centre of the TV screen, and at a height of 50 inches from the ground. A PC was running a bespoke MAX MSP patch to convert the 5.1 output of the PS4 into a standard stereo downmix and a binaural stereo downmix with a switch between. Figure 2 shows the patch UI and components of the process. The virtualiser is a key component to the binaural process as it replicates the 5.1 listening experience over headphones.

For this experiment HRTFs from a KEMAR binaural mannequin were used [6]. Room modeling reverberation using IRCAM's Spat reverberator was used to aid externalization and give a more convincing virtual loudspeaker listening experience. The reverberation time was set to 0.3s, which matched the RT60 of the laboratory A panner patch was also used to make sure the calibrations of the downmixer were correct and each speaker direction matched the signal panning as shown in Figure 3. The downmixer used standard coefficients when panning the surround and LFE channels to the left and right channel. The LFE was reduced by 10dB and sent to all channels equally. The left surround and right surround were reduced by 6dB, and the centre channel reduced by 3dB.

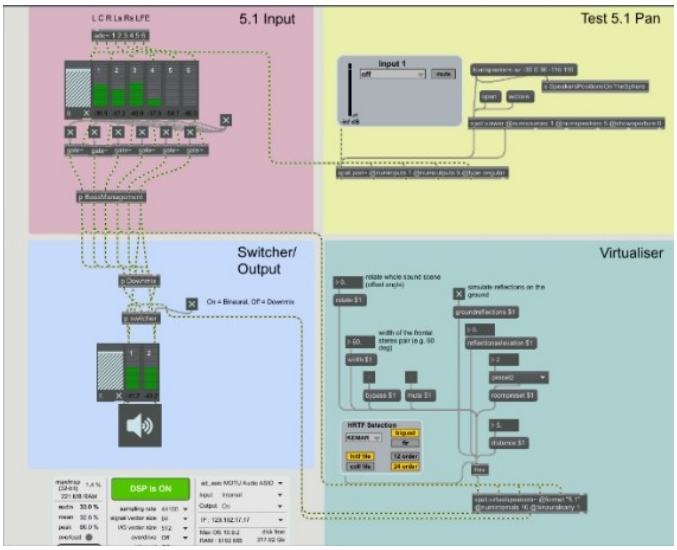


Figure 2: Binaural and Stereo Downmix Max MSP patch

Participants were given a consent form to sign, followed by a preparation sheet to inform them of the experiment requirements, conditions, game context and controls. 22 participants were asked to play a specific 5 minute segment of the game with the lights off. 11 participants played the game listening to standard headphone stereo, while the other 11 played with the 5.1 binaural playback system on. After the participants had played the game for 5 minutes, they were then asked to complete the Immersive Experience Questionnaire (IEQ), mentioned earlier [10], to assess their levels of immersion throughout the experience.

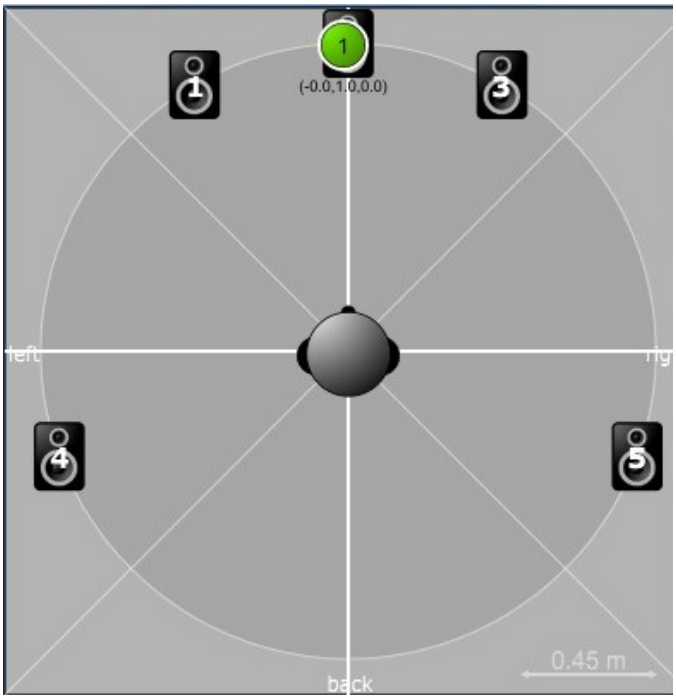


Figure 3: Binaural and Stereo Downmix Max MSP patch

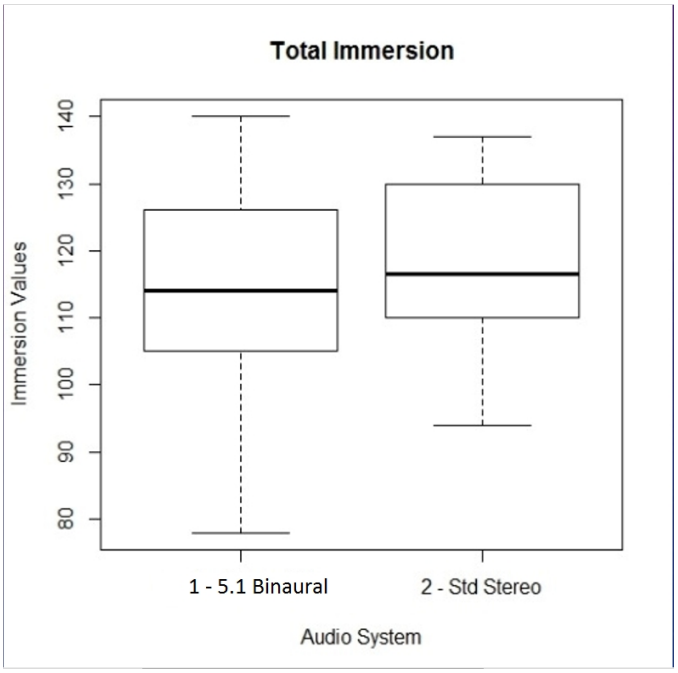


Figure 4: Total Immersion Boxplot

4. RESULTS

As the data was normally distributed and between two groups, T-tests were used for the statistical analysis. The test was gender balanced with 11 males and 11 females, with varying age groups. This consisted of 1 person in the 17-21 category, 7 in the 22-25 category, 5 in the 25-30 range and 9 in the 31+ category.

As seen in Figure 4 there is no significant difference in the means between players who listened to the binaural stereo soundtrack and players that listened to the standard stereo soundtrack. A t-test was conducted on the Total Immersion results, with **t** (df) = -0.46833 and **p** = 0.6446. After analysing the components of the questionaire in boxplot form, only two components appear to show any particular difference between the groups. These are the cognitive component and the challenge component.

In Figures 5 and 6, it appears there is a difference to make note of visually, however they are opposite in terms of which audio setup has a higher value. It appears that with regards to cognitive involvement, it could be argued that standard stereo playback achieves a higher value than binaural stereo. This therefore supports the hypothesis but not in terms of the direction proposed.

With the challenge component, it could be argued to support the claim that binaural stereo has a higher level of immersion than standard stereo playback. Further analysis of the results via a t-test show that the difference in the challenge component is not statistically significant, with **t** (df) = 1.747 with a **p** of 0.09599.

5. DISCUSSION

The results in this experiment do not support the hypothesis stated earlier; that the effect of 5.1 binaural audio on levels of immersion will be significantly different than the effects of standard stereo. The assumption that 5.1 binaural audio playback will increase levels of immersion was also not supported. In the case of cognitive involvement it could be argued that standard stereo caused a higher level of immersion. There are many reasons that could contribute to the state of these results that indicate the necessity of further investigation.

Although the boxplots displayed in analysing the Total Immersion and its components indicate there may be some difference between groups in specific areas, further statistical analysis showed that this was not the case. The values of **t** (df) and **p** in the statistical analysis support the null hypothesis that the difference between both categories is not statistically significantly. However, these results suggest that a larger participant group could lean towards a more significant value, especially in the challenge area. Despite there being 22 participants in this pilot study, more data is required to generalise the results so that they can be made applicable to the general population. The use of binaural technology in gaming is still relatively new, and this also could cause those unaccustomed to binaural rendering to become distracted if the rendering is not effective. A further experiment may involve the participant's own measured HRTFs and a phase of becoming accustomed to the rendering before the test. Potentially, this may be why the cognitive involvement was lower and the challenge level was higher in the binaural results. The adaptation process to the technology may have cause the player to have more difficulty when playing the game as opposed to the system increasing levels of immersion. It should be noted however that individualised HRTF measurement is time consuming, complex and expensive, requiring large, specific equipment to achieve and is not representative of the current consumption of binaurally rendered material.

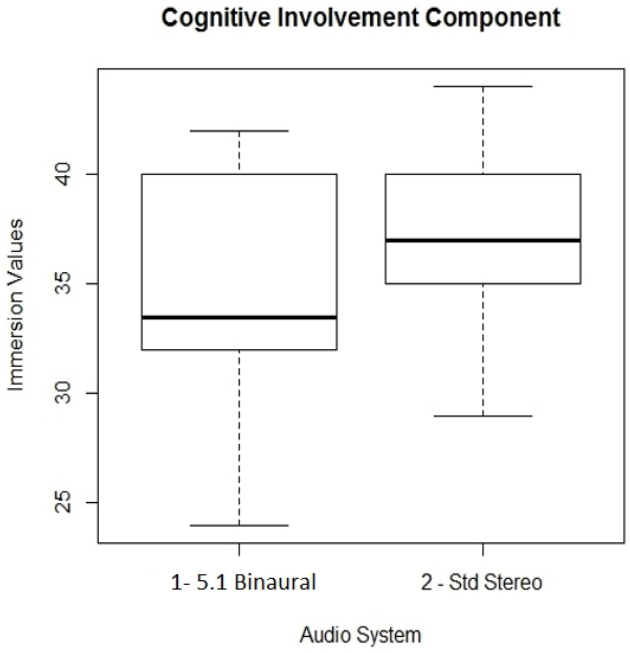


Figure 5: Cognitive Involvement Boxplot

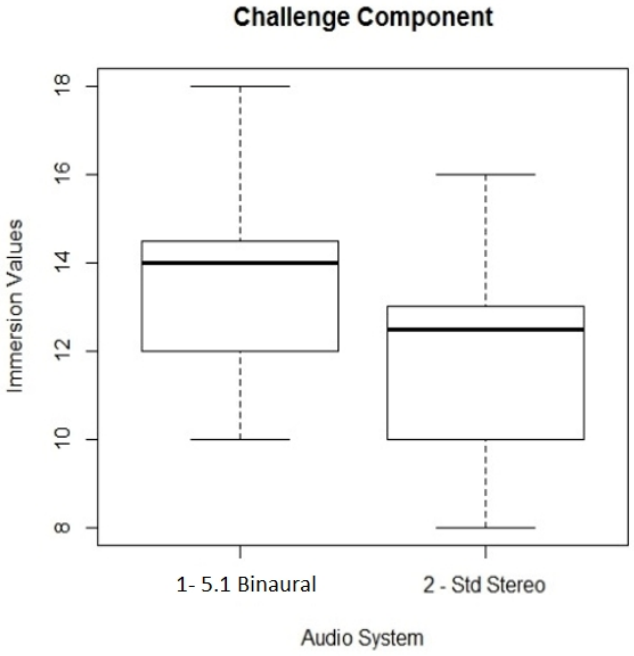


Figure 6: Cognitive Involvement Boxplot

The use of a TV screen is also not very immersive visually. It would also be of interest to utilize more immersive visual displays in addition to the 3D audio system. A VR system that incorporated head-tracking would be a highly suited piece of equipment to use in further work. Not only would it create a much more immersive visual experience, the head tracking data could also be used to manipulate the audio soundtrack to replicate our real world auditory experience in a much more detailed manner. Similarly to the equipment, the player is new to the soundtrack and the mechanics of the game. If the player is allowed to grow accustomed to these, they may be more attuned to discriminate differences in immersion.

Although this study appears to support the null hypothesis, the process of binaural virtualization must also be considered. The binaural virtualization is only on the azimuthal plane and is therefore not a true representation of what we experience in real life. To further explore the effects of binaural soundtracks, we must also fully replicate a 3D audio experience that is not only inclusive of one plane.

6. CONCLUSION

The results presented in this paper have shown evidence that 5.1 binaural audio does not improve the player's sense of immersion over standard stereo rendering under the test conditions reported. However they do imply a connection between binaural audio and immersion. To ascertain this connection it would be beneficial to do further experiments, that incorporate a full binaural rendering of the soundfield and not just 5.1. Using VR equipment, head-tracking data and 3D virtualisation with specifically measured HRTFs would be the most appropriate way to replicate a 'realistic' gaming experience, which can be used to further assess levels of player engagement and immersion.

7. ACKNOWLEDGMENTS

This research is supported by the Engineering and Physical Sciences Research Council (EPSRC) and the EPSRC Doctoral Research Centre in Intelligent Games and Games Intelligence. (IGGI).

8. REFERENCES

[1] D. R. Begault, E. M. Wenzel, and M. R. Anderson. Direct comparison of the impact of head tracking, reverberation, and individualized head-related transfer functions on the spatial perception of a virtual speech source. *Journal of the Audio Engineering Society*, 49(10):904–916, 2001.

[2] J. H. Brockmyer, C. M. Fox, K. A. Curtiss, E. McBroom, K. M. Burkhart, and J. N. Pidruzny. The development of the game engagement questionnaire: A measure of engagement in video game-playing. *Journal of Experimental Social Psychology*, 45(4):624–634, 2009.

[3] G. Cassidy and R. Macdonald. The effects of music choice on task performance: A study of the impact of self-selected and experimenter-selected music on driving game performance and experience. *Musicae Scientiae*, 13(2):357–386, 2009.

[4] C. I. Cheng and G. H. Wakefield. Introduction to head-related transfer functions (hrtfs): Representations of hrtfs in time, frequency, and space. In *Audio Engineering Society Convention 107*. Audio Engineering Society, 1999.

[5] Dolby. Dolby atmos audio technology. http://www.dolby.com/us/en/brands/dolby-atmos.html. (Accessed on 09/09/2016).

[6] GRAS. Gras sound & vibration. http://kemar.us/. (Accessed on 09/09/2016).

[7] S. Hughes and G. Kearney. Fear and localisation: Emotional fine-tuning utlising multiple source directions. In *Audio Engineering Society Conference: 56th International Conference: Audio for Games*. Audio Engineering Society, 2015.

[8] S. Huiberts. *Captivating sound the role of audio for immersion in computer games*. PhD thesis, University of Portsmouth, 2010.

[9] G. Interactive. Doom. Windows, 1994.

[10] C. Jennett, A. L. Cox, P. Cairns, S. Dhoparee, A. Epps, T. Tijs, and A. Walton. Measuring and defining the experience of immersion in games. *International journal of human-computer studies*, 66(9):641–661, 2008.

[11] NAMCO. Ridge racer v. PS2, 2000.

[12] NINTENDO. The legend of zelda: Twilight princess. Wii, 2006.

[13] F. Rumsey. *Spatial audio*. CRC Press, 2012.

[14] SEGA. Alien: Isolation. PS4, 2014.

[15] M. Slater and S. Wilbur. A framework for immersive virtual environments (five): Speculations on the role of presence in virtual environments. *Presence: Teleoperators and virtual environments*, 6(6):603–616, 1997.

[16] R. J. Tafalla. Gender differences in cardiovascular reactivity and game performance related to sensory modality in violent video game play1. *Journal of Applied Social Psychology*, 37(9):2008–2023, 2007.

[17] S.-L. Tan, J. Baxa, and M. P. Spackman. Effects of built-in audio versus unrelated background music on performance in an adventure role-playing game. *International Journal of Gaming and Computer-Mediated Simulations*, 2:142–64, 2012.

[18] A. Technologies. Auro-3d / auro technologies : Three-dimensional sound. http://www.auro-3d.com/. (Accessed on 09/09/2016).

[19] D. Wang and G. J. Brown. *Computational auditory scene analysis: Principles, algorithms, and applications*. Wiley-IEEE Press, 2006.

[20] M. Yamada, N. Fujisawa, and S. Komori. The effect of music on the performance and impression in a video racing game. *Journal of Music Perception and Cognition*, 7(2):65–76, 2001.

Investigating the Impact of Source Spectra on Spatialized Audio Content

Sally-anne Kellaway

University of Sydney
Sydney, Australia
+61 423355368
s.a.kellaway@gmail.com

ABSTRACT

It is widely accepted that the ear imparts a filter on incoming sound signals in order to understand the position of a sound source in relation to the listener. The characterisation of this filter is the Head Related Transfer Function (HRTF), which has decades of research history investigating the impact of the various spectral elements on the human ability to localise sound.

As audio professionals, we also understand that spectrums are able to be summed and manipulated in various ways that create new spectra. One way that this can manifest is with the summing of the spectra of source sound, HRTF spectra, and the delivery hardware.

The speaker reports on research findings that demonstrate evidence for this effect having a prevalent effect on delivery of HRTF processed audio in Virtual Audio Displays (VADs). This presentation will focus on the impact of source spectra on HRTF processing, as an example of these interactions. Investigating these impacts by referencing research conducted in laboratory conditions, we are able to extrapolate and discuss the potential impacts of this effect in industry. Attendees will leave this session with an understanding of the impact of elements of their pipeline on the effect of HRTF processing in their Virtual/Augmented and Mixed Reality projects, including source and delivery mediums.

CCS Concepts

Applied computing → Arts and humanities → Sound and music computing

Keywords

Virtual reality; augmented reality; binaural; localization; spectral colouration; virtual auditory display;

1.INTRODUCTION

An interesting distinction raised within Kistler and Wightman's 1997 study was that while spectral uncertainty negatively influences localization ability, 'normal' listening conditions with 'normal' sounds do not present as constant, consistent spectra. Any number of variations can occur with the source or environment that may create interferences with the spectra arriving at the ear. This introduces the role of experimentation with systematic variations in spectra within localization tasks. To understand how these 'normal' variations impact on the human ability to localise sounds, it is necessary to understand how the interaction between the Head Related Transfer Function and the Spectra of the Source sound.

It is widely accepted that the ear imparts a filter on incoming sound signals in order to understand the position of a sound source in relation to the listener. The characterisation of this filter is the Head Related Transfer Function (HRTF), which has decades of research history investigating the impact of the various individual and combined spectral elements on the human ability to localise sound.

It is important to note that this study calibrates its context to the point beyond the duplex theory. It is well acknowledged by many, and well tested, documented and discussed (including Kistler and Wightman (1997)) that accurate localization is achieved by the use of monoaural spectral cues. It is recognized that ITD and ILD do have an impact on localization to some extent (particularly away from the 'cone of confusion' regions), but research such as Kistler and Wightman (1997) conclusively demonstrate that if one ear is plugged with an ear plug and covered with a muting cover, localization is still possible.

The specific spectral cues for front-back directional distinctions are well documented (see, for example, Wightman and Kistler, 1997; Zahorik, Bangayan, Sundareswaran, Wang & Tam, 2006; Sunder, et. al., 2012). Furthermore, there is extensive consensus that there lies a significant challenge in achieving consistent and accurate listener identification of front and rear sources, when listening via headphones (when no head tracking is used) (Zahorik, Bangayan, Sundareswaran, Wang, & Tam, 2006) (Sunder, 2012), (Begault, 1990), (Kelly, 2003). Recognizing the unacceptably high rate of front-back reversals, each author presents solutions to this challenge; however, considering the impracticality of individualized HRTF measurement (when considering the mass appeal of the application of such technology such as Virtual Reality Video Gaming), there exists the need to look beyond these high-requirement solutions, and to offer a non-individualised option.

However, it is also clear from the rate of development of software processing technology available in the virtual reality audio processing software field that a non-individualised solution may not be readily available in the near future. Until non-individualised solutions are available, it may be beneficial for practitioners to understand the impact of source colouration on human localisation ability in virtual auditory fields.

2. THE ROLE OF SOURCE SPECTRA

Hebrank and Wright (1974) touch on a consensus in their research, that when studying the role of spectral cues in directional hearing that white noise is the best "vessel" for these tests. From the earliest studies (e.g., (Butler, 1968), (Blauert, 1969)), and even up to current listening experiments, the practice of using short bursts of white noise has formed a salient research methodology.

This prompts us to recognize that there are two perspectives when investigating the role of spectra in localisation. One is the investigation of the human element of the system – the pinna system and subsequent interpretation of localization cues from this system. This is the perspective we investigate from when considering the role of HRTFs in localisation. The other perspective is from the source spectra, and the way by which this element can impact and colour on perception and localization.

2.1 Traditional Approaches to Understanding Localisation

A more traditional perspective on spatial analysis of spectra influence is achieved from the angle of spectral analysis of spatially captured data (by capture of a Head Related Transfer Function (HRTF)). This style of analysis has delivered a wealth of data that correlates to deliver spectral cues that are critical for accurate perception of auditory source direction. Even when considering one singular discrimination – Front to Back – there is wide range of documentation - see, for example, Kistler and Wightman (1997) or Zahorik, Bangayan, Sundareswaran, Wang, and Tam (2006) document the importance of the frequency bands 1kHz for Rearward and 3.5kHz for Frontal sources.

This style of analysis does in fact contribute to this study – specifically denoting the importance of spectral bands for certain localisation tasks. Identifying them as features of a listener's HRTF also raises the question that there might be an impact to the accuracy of localisation if these discrete cues are manipulated. This effect was documented in the preceding study, and will be continually investigated in the current study.

2.2 Spectral-Directional Colouring and Bias

It is critical to understand the influence of discrete cues, and to consider the extreme spectral bias that can be imparted by such cues. This is often referred to as spectral-directional bias in research.

One such theory is the Covert Peak Area theory. Butler's contributions on Covert Peak Areas (CPAs) were proposed in 1984 and elaborated through a series of studies in 1987, 1988, 1990 and 1992. CPAs are explained by Butler as the place in space where one narrow frequency band (or segment) of the recorded sound spectrum is amplified more than it is from any other spatial location. Butler characterised a "more amplified band" as a 1kHz-wide band having a recorded magnitude near maximum levels (Rogers & Butler, 1992).

These peaks are identified as 'covert' because they are not immediately clear when magnitude is plotted against frequency. Only when the dB gain is plotted against the location of the sound source do these 'covert' peaks appear across the areas measured.

These CPAs are not just measured peaks; they are also detectable and localisable features. Butler and Rogers delve deeply into extracting CPAs for the frontal hemisphere, identifying that the CPAs in this area more adequately account for monaural localization performance of Vertical-Plane positioned sounds (Butler et al., 1990). Measured results indicate that there are an abundance of CPAs to allow for localization across the extremes of the coronal plane.

Studies focussing on the identification of spectral-directional colouring and bias for elevated locations commenced with Blauert's first study in 1969. Blauert proposed that with a fixed head position, the dominating frequency of the source sound has a significant impact on localization. Blauert was able to characterise both "directional" and "boosted" bands by playing narrow-based noise to participants. It was then concluded that "directional bands" in the binaural signals enabled the discrimination of the direction of Frontward, Rearward and Elevated (above ear-level) sources across the total pool of participants.

While there are results presented include elevation, Blauert focuses on Front to Rear differences in his 1969 study. A more detailed characterisation on the horizontal planes can be delivered by delving in to more recent studies, such as Martens' 1987 study on spatial energy.

Martens (1987) developed a series of detailed spatial energy plots that demonstrate the inherent spatial qualities of various frequency bands, and used the same measurements to calculate localisation ability as a function of two Principal Components. Most significantly, these measurements are subject to a principal components analysis (PCA) to determine two "rotational weighting" filters that are applicable (to different degrees) to each location on the 360-degree azimuthal plane.

The data from one participant in the study (MDL) was published in detail, as the PCA results from MDL were highly applicable to the other subjects in the study. To note the most significant details for this study, at frontward locations (0 degrees), the 250Hz and 4kHz bands become positively weighted and correlate with Blauert's directional bands. Rearward cues that were features in Blauert's directional bands (1kHz particularly), are up-weighted by the PC score to feature a significant peak at 1kHz.

For specific studies on spectral-directional bias such as these studies from Blauert, Butler and Martens, it can be considered that these are three presentations of similar datasets. All three are spatially varying visualizations of magnitude as a function of frequency. They all demonstrate clear correlations between each dataset, which describes a pattern of spatial movement across the frequency spectrum.

It is also possible to detect a change in localization ability across the frequency spectrum. When discussing King and Oldfield's 1997 findings, Letowski and Letowski (2012) discuss the concept of a "center of gravity" in the frequency spectrum of the source. This effect describes a (frequency) point at which the difference in accuracy in localizing high and low frequency sounds is at a peak (maximum difference in accuracy). This centre of gravity theory aligns with previous discussions of spectral-directional bias because the stimulus sounds that are used in the experiment are highly resonant, formant-based vocal vowel sounds. These sounds do have very clear differences in tone colour, and hence each have a strong "centre of gravity" where the majority of the spectral energy is focused (the first formant is the most noticeable peak of magnitudinal energy).

3. ANALYSIS OF SOURCE IMPACT

This deep consideration of several theories on spectral-directional colouration is critical to understanding the impact of spectra on human localisation. Understanding that spatial locations may impart severe spectral colouration on incoming sound prepares us to consider the impact of source spectra.

In the same way that Butler was able to identify Covert Peaks that influenced listener perception of location, the spectral filters of certain sound sources can be described as exaggerated enough to 'imply' a location other than the sound sources actual location. Within certain contexts, especially in research such as this, the impact of certain elements of the spectra are investigated not as arbitrary interferences, but as systematic variations. Systematic variations can help to characterize targeted elements that impact on localization, and by their nature may be more salient to replicated experimentation across many experiment condition (through the virtue of reproducibility).

In order to identify a systematic variation to a possible listener HRTF, and effect to the ability of a human listener to localise sound, this study will present a visual analysis of a HRTF, vocal vowel spectra and the combined result of these.

3.1 Method for Graphing Spectra Influence

For the collection of the 40 participant Head Related Impulse Responses (HRIR), a chair was set for each listener in a near-anechoic room at the University of Sydney, where participants were asked to block their ear canals with Anti-Nois™ earplug wax to support a pair of B&K Type 4101 miniature ear canal microphones. This is consistent with previous headphone listening studies such as Møller et al. (1995). To derive data for HRIR measurements, a swept logarithmic sine tone was played from two Yamaha MSP3s that were placed at front and rear positions with azimuth angles of approximately 40° and 140°, and at a distance of approximately 1.5 meters from the centre of the subject's head. This style of HRIR capture is equivalent to the methods outlined in Farina's research (2000).

To prepare for graphing, the databases were subject to time alignment to extract the Head Shadowing component of the impulse. The data was then converted to frequency domain and the mean of the total subject pool response was calculated. This mean is then divided out to isolate variances from the average response, hence revealing direction-varying elements. This averaging of the means is modelled after the data preparation that Martens executed in his 1987 study (Martens, 1987).

In a follow up study, a practical experiment will formally explore the impact of "A", "U" or "I" vowel sound spectra convolved with either the frontal or rear HRTF. The spectra for these three vowel sounds are presented here as examples of highly resonant potential source sounds.

3.2 HRTF Spectra

Mean magnitude across 40 participants is charted between 100Hz and 10,000Hz. For each Impulse Response played to the listener, both the ipsilateral and contralateral ears are captures and charted. It is worthwhile noting some of the critical bands that were covered earlier in this document - 1kHz for rearward sources and 3.5kHz for frontward sources are consistent even in this dataset.

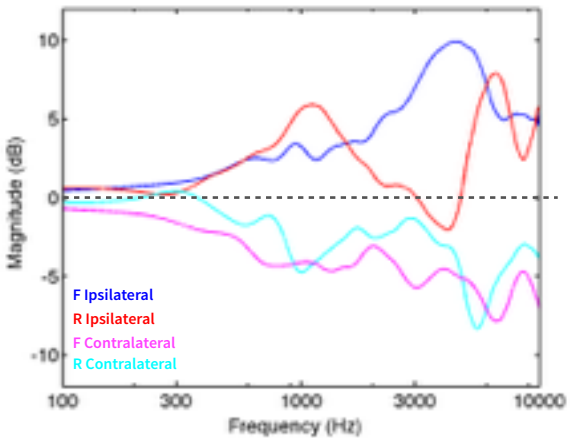


Fig 1. Magnitude between 100Hz and 10,000 Hz, leaned across 40 participants for 2 locations (40 degrees and 140 degrees)

3.2 Vocal Vowel Source Spectra

'Source' Spectra of three vocal vowel sounds are charted between 200Hz and 8,000Hz. For the purpose of clarity, each vowel spectrum was subject to a long-time-average 50-pole LPC analysis to show the resonant structure of the vowels. This allows the major features (formants) of the vowels to be visually compared without the fine spectral detail that may be a result of periodic glottal modulation.

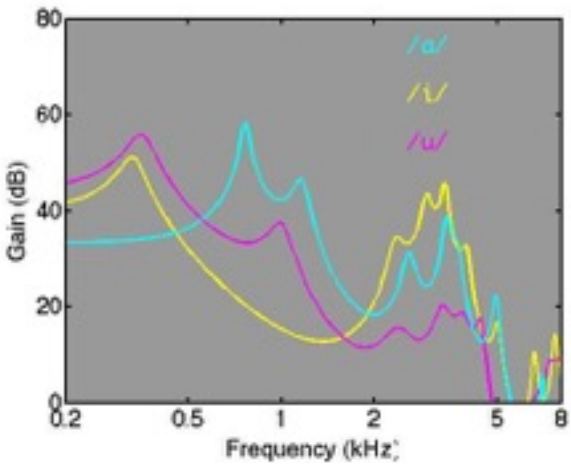


Fig 2. Magnitude between 100Hz and 10,000 Hz, leaned across 40 participants for 2 locations (40 degrees and 140 degrees)

3.3 Vocal Vowel Source Spectra Convolved with HRTF spectra

As audio professionals, we understand that spectrums are able to be summed and manipulated in various ways that create new spectra. One way that this can manifest is with the summing of the spectra of source sound, HRTF spectra, and the delivery hardware.

To consider the impact of the source sound spectra, we combine the spectra of the vocal vowel sounds selected with the mean HRTF of the participant pool to visually inspect the difference in original and impacted source spectra. To consider two types of potential spectrum manipulation, the spectrum for "I" and for "U" are convolved with the meaned HRTF data.

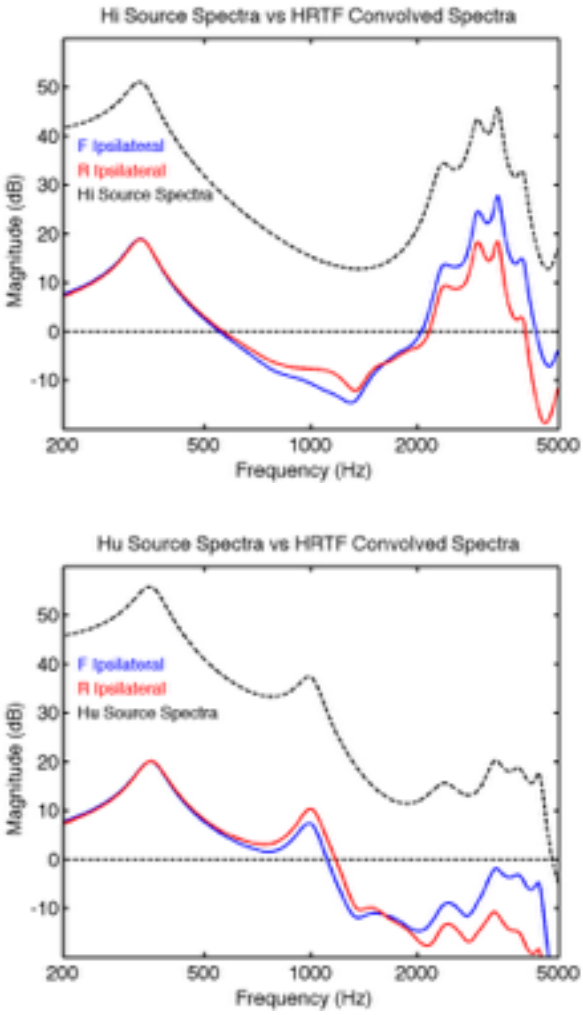


Fig 3. Magnitude between 100Hz and 10,000 Hz, leaned across 40 participants for 2 locations (40 degrees and 140 degrees). Top panel is the spectra for "I" convolved with the leaned HRTF data and the bottom panel is the spectra for "U" convolved with the leaned HRTF data.

4.DISCUSSION

Visual analysis reveals a clear correlation between the source spectra and critical cues in the HRTFs presented. This can be highlighted by pointing out a number of correlations: spectra bands with Blauert's Directional Bands, convolved spectra with HRTF cues, and considering the vowel spectra against other spectral manipulation research.

4.1 Vowel Spectra and Spectral-Directionality

When assessing the potential impact to human localisation ability, it is valid to consider the alignment of the vowel spectra with spectral-directional features discussed by authors such as Blauert. In his 1969 research, Blauert identified boosted frequency bands that correlate with specific localisable areas. Considering the vowel spectra of the "I" and "U" cases in Figure 2, we can easily see the major peaks of the "I" spectra at the 3.5kHz point, which correlates with the frontward directional band shown below.

This correlation carries across to the "U" spectra as well, where we can easily see a boost in the 1kHz band that correlates with the major rearward directional band shows in Figure 4 below. Generally speaking, these observations are salient with Butler's CPA theory as well (1992), where the results of that study found that generally "high" spectral ranges of the source presented to the monaural listeners were found to be be localised more frontward and elevated than generally "low" spectral ranges of the source sounds.

It should be noted that the "A" spectra, while it also demonstrates the 1kHz band boost, also demonstrates the frontward band boost at 3.5kHz as well. This effect is hypothesised to "cancel out" any particular bias in the locations that were used for the HRTF capture (40 degrees and 140 degrees).

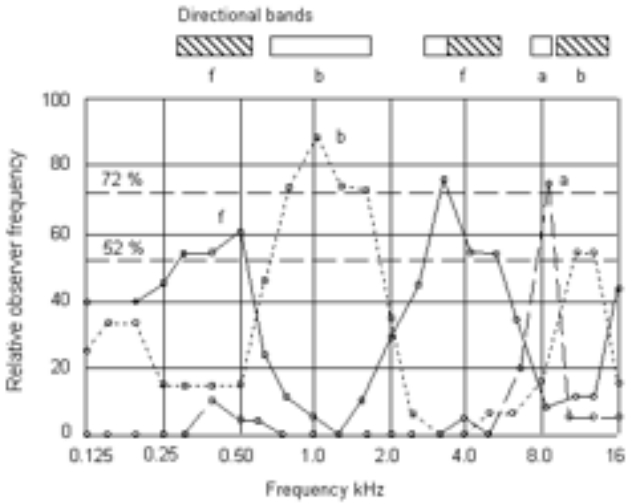


Fig 4. Blauert's directional bands (1969), where f = frontward, b = rearward and a = elevated. The bands shown are percentage of observer frequency of identification of that direction plotted against (sound) frequency.

4.2 HRTF Manipulation and Augmentation

Visually comparing the spectra of the HRTFs recorded for the subject pool to both the source spectra and the convolved spectra, we can easily identify manipulations to the major Frontward (appx 3.5kHz) and rearward (appx 1kHz) cues.

For the "I" spectra, it is possible to note an augmentation of the 3.5kHz spectral range of approximately 10dB. A similar difference is notable between the "I" spectrum convolved with the Frontward and Rearward HRTFs.

We can also notice a similar effect between the convolved "U" spectrum and the Rearward position HRTF. Again, we see an augmentation of the Rearward (1kHz) cue in the convolved spectrum, but this time the difference between the Frontward and Rearward convolved HRTFs at that 1kHz cue is not as pronounced. However, there is a significant augmentation in the Frontward cue area (3kHz+) that would assist with frontward localisation. It is also worthwhile to note, though, that the degradation of the Frontward HRTF cues in the "U" is extreme.

5.CONCLUSION

This HRTF cue degradation and augmentation from vocal vowel spectra has serious implications for creative practitioners working in virtual and augmented reality sound and music design fields. This visual analysis focussed on the resonance of vocal vowels, and has demonstrated the way by which the summation of HTRF spectra and source sound spectra from vocal vowel sounds could

alter the localisation cues that are necessary for Frontward and Rearward localisation.

Hypothesising on the impact of spectral degradation and augmentation of HRTF cues for creative professionals in the VR and AR, it is feasible to suggest that despite our best efforts to perfect spatial sound positioning, that the spectral colourations introduced by source spectra (and possibly end medium colouration as well) may defeat exacted spatial designs. To remedy this, it would be interesting experiment to test the impact of purposefully colouring Frontward and Rearward sound sources on a global level to augment their spatial positioning.

6.ACKNOWLEDGMENTS

The author would like to acknowledge the assistance of Luis Miranda and Manuj Yadav for organizing and setting up the listening experiment, and William Martens for providing the introductory MATLAB script from which the analysis of the results and generation of figures was derived.

7.REFERENCES

1. Begault, Durand. (1990). Challenges to the Successful Implementation of 3-D Sound. Paper presented at the 89th AES Convention, Los Angeles USA.
2. Blauert, J. (1969). Sound Localization in the Median Plane (Vol. 22, pp. 205-213). Acustica.
3. Butler, Suzanne K. Roffler; Robert A. (1968). Factors that Influence the Localisation of Sound in the Vertical Plane. The Journal of the Acoustical Society of America, 45(6), 4.
4. Butler, R. A., Humanski, R. A., & Musicant, A. D. (1990). Binaural and Monaural Localisation of Sound in Two-dimensional Space. *Perception, 19*(2), 16.
5. Farina, A. (2000, February 19-22). Simultaneous Measurement of Impulse Response and Distortion with a Swept Sine Technique. Paper presented at the 108th AES Convention, Paris, France.
6. Hebrank, J., & Wright, D. (1974). Spectral Cues Used in the Localisation of a Sources on the Median Plane. *Journal of the Acoustical Society of America, 56*(6), 1829-1834.
7. Kelly, Michael C.; Tew, Anthony I. (2003, 22-25 March). A Novel Method for the Efficient Comparison of Spatialisation Conditions. Paper presented at the 114th AES Convention, Amsterdam, DE.
8. Kistler, D. J., & Wightman, F. L. (1997). Monaural Sound Localisation Revisited. *Journal of the Acoustical Society of America, 101*(2), 13.
9. Letowski, T., & Letowski, S. (2012). Auditory Spatial Perception: Auditory Localization: US Army Research Laboratory.
10. Martens, W. L. (1987). Principal Components Analysis and Resynthesis of Spectral Cues to Percieved Direction. *ICMC Proceedings*, 274-281.
11. Møller, H; Hammershøi, D; Jensen, C; & Sørensen, M. (1995). Transfer Chracteristics of Headphones Measured on Human Ears. Journal of the Audio Engineering Society, 43(4), 14.
12. Rogers, M. E., & Butler, R. A. (1992). The Linkage Between Stimulus Frequency and Covert Peak Areas as it Related to Monaural Localisation. *Perception & Psychophysics, 52*(2), 10.
13. Sunder, Kaushik; Tan, Ee-Leng; Gan, Woon-Seng. (2012). On the Study of Frontal-Emitter Headphone to Improve 3-D Audio Playback. Paper presented at the 133rd AES Convention, San Francisco, USA.
14. Zahorik, P., Bangayan, P., Sundareswaran, V., Wang, K., & Tam, C. (2006). Perceptual recalibration in human sound localization: Learning to remediate front-back reversals. *The Journal of the Acoustical Society of America, 120*(1), 343. doi: 10.1121/1.2208429

Some Possibilities for Cellular Automata in Game Audio

Isaac Schankler
California State Polytechnic University, Pomona
Music Department
Pomona, CA 91768
+1 (909) 979-3564
ischankler@cpp.edu

ABSTRACT
As a method to create complex patterns from simple rules, cellular automata (CA) have long appealed to composers and musicians as tools for musical generation. In the realms of academic research and experimental music, CA-generated music is a well-worn topic. In the world of game audio, however, it remains an almost entirely unexplored technique. In this paper I will discuss some of the logistical and aesthetic considerations when using CA for musical generation, some advantages of using CA for game audio, and some limitations and challenges that still exist.

Keywords
Cellular automata, totalistic cellular automata, generative music, game audio, procedural audio

1. INTRODUCTION
Cellular automata (CA), as a way to create complex patterns from simple rules, have long been objects of fascination for mathematicians, scientists, and artists alike. Many composers and musicians have created music and sound art based on CA. Iannis Xenakis, in his 1986 piece for orchestra *Horos*, used CA to determine the orchestration and pitch content of certain musical material [6]. Other composers who have worked with CA include Peter Adriaansz, Peter Beyls, David Burraston (Noyzelab), Paul Hembree, Eduardo Miranda, and Carla Scaletti, though this is by no means a comprehensive list [2,5]. Most explorations of CA to date have been in the realm of concert music or academic research; it remains a relatively unexplored topic in game audio, despite recently renewed interest in procedural audio for games.

1.1. Humanizing Automata
Some musicians and artists use methods to complicate or "humanize" the mechanical output of CA, in order to make them more suitable or flexible for artistic purposes. Christopher Ariza uses probabilistic methods to "bend" the musical output of CA [1]. In the world of visual art, Gwen Fisher's *Pixel Paintings* hover between abstract and representational art, incorporating intuitively selected colors and a handmade aesthetic influenced by folk art traditions (see Figure 1) [3]. This provides a potential model or goal for how CA might eventually function in game audio; while mathematically driven, CA can be worked with and used intuitively by composers, just like any other raw musical material.

2. AESTHETIC CONSIDERATIONS
Relative to more familiar musical materials, CA do present unique aesthetic considerations for the composer. CA begin with a row or grid of *cells* in a certain number of dimensions (usually 1d or 2d). Each cell has a finite number of possible *states* or *colors*. Each turn (or time-step), cells change their states based on a predetermined *rule.* Generally, cells will only be affected by cells that are close by, and this area is called a *range* or *neighborhood*.

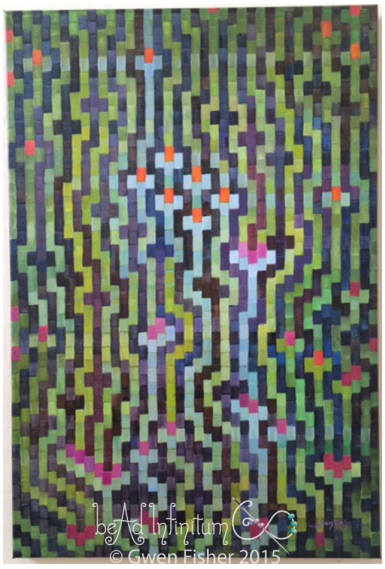


Figure 1. One of Gwen Fisher's *Pixel Paintings*

2.1. Randomness and Complexity
Not all CA generate complex results; Stephen Wolfram [7] has attempted to qualitatively classify automata based on their relative complexity. A Class 1 automaton evolves to a homogenous state; Class 2 to a stable or periodic state; Class 3 to a chaotic pattern; and Class 4 to complex localized structures. But so far, no one has been able to define these classifications in a mathematically rigorous way, and the boundaries between them can be fuzzy or subjective, especially between Class 3 and Class 4. At least for now, these classifications seem to say just as much about human perception as they do about the automata themselves.

For example, elementary cellular automaton Rule 30 is considered Class 3. Rule 30 passes several statistical tests for randomness, and is often used as a pseudo-random number generator [4]. However, it is visually distinguishable from randomness; for example, in the propagation of the automaton from a single cell, a repeating figure is visible along the left edge of the pattern, and triangle-like shapes of varying sizes are visible throughout the pattern (Figure 2). Conversely, zooming out far enough makes the pattern indistinguishable from white noise to the human eye.

Thus, when discussing CA, we must make a distinction between complexity and randomness. Ideally, we would like to bring out those features of an automaton that seem distinct from randomness. This is a fuzzy thing to define and identify. What do we mean when we say we see a pattern, and when we say something appears random? Randomness alone is not musically interesting in and of itself. If we are to find musically interesting ways to use cellular automata, we must focus on the

characteristics that seem to defy randomness, that seem indicative of meaningful patterns. This is not always a straightforward task.

Figure 2. Propagation of Rule 30 from a single cell

2.2. Temporality

The temporality or linearity of music presents an additional challenge to musical coherence. When observing the visual propagation of a 1-dimensional automaton, for example, we have a certain level of agency, in that our eyes are free to roam around, to compare the state of the automaton at various stages, and to hunt for larger patterns in that propagation. But when this propagation is sonified, we are bound to perceive the state of the automaton linearly, and it becomes more difficult to compare states and detect patterns, especially with states that are far apart.

What we lose in terms of visual agency is offset somewhat by our auditory abilities; for example, our ability to separate sound sources, and to hear counterpoint. This allows us to, among other things, listen to the output of multiple cellular automata simultaneously.

Thus there are a variety of factors to consider when generating musical material from cellular automata. What kind of automaton, what rule to choose, what boundary conditions and initial state to use, and how to convert the information into sound can all have dramatic implications for the music generated by the automaton. In creating our musical generation system, we must make informed choices (or at least educated guesses) about these factors to set us up for a musically effective and compelling result.

3. AESTHETIC DECISIONS

3.1. Dimensionality

The first, most fundamental question to ask is what type of automaton to use. Conway's Game of Life, a 2-dimensional totalistic cellular automaton, is a popular automaton for musical generation. In many of these automaton-to-sound mappings, the x-axis corresponds to time, and the y-axis corresponds to pitch. While this is an obvious mapping for anyone who has used a sequencer or digital audio workstation, it is not a particularly musically meaningful mapping. In the Game of Life and other 2-dimensional CA, the x-axis and y-axis are functionally identical. There is nothing intrinsic about 2-dimensional automata that would cause one axis to be mapped to time and another to pitch. While it is possible to create interesting sonic patterns with this method, it is not especially intuitive to explore, and it often tends to generate musical patterns that sound arbitrary or meandering.

By restricting the automaton to 1 dimension, we can have this dimension deal with time exclusively. Furthermore, we can subdivide time in a hierarchical way, with each row being equivalent to one measure, and each cell corresponding to a particular time-point in that measure. This hierarchical division of time mimics the way we perceive music.

3.2. Colors

Many decisions about what kind of automaton to use are a kind of dance between creating a set of automata with a variety of musical possibilities, while keeping those possibilities to a manageable number that is not completely overwhelming to explore. One such choice is determining the number of colors or states to use. Elementary cellular automata are binary; that is, they only use two colors: black and white, on and off, 0 and 1. There are only 256 possible rules for elementary cellular automata.

While this is a very manageable space to explore, it is somewhat limited in terms of musical possibilities. Most music hinges on parameters that are distinctly non-binary; that is, qualities that exist on a continuum or spectrum, like pitch and dynamics. Thus it makes sense to use an automaton with several colors that can be mapped to points along such a continuum. For purposes of this paper, we will use an automaton with 5 colors that can be mapped to, for example, a pentatonic scale, or 5 distinct dynamic levels. This is not a magic number, but it is perhaps the minimum number that will give us sufficient expressiveness.

3.3. Totalistic vs. Non-totalistic

However, just increasing the number of colors from 2 to 5 dramatically increases the number of possible rules to 5^{125}, or about 4.2×10^{37}. One way to make this space more manageable is to use totalistic cellular automata. Instead of creating rules based on the content of each individual cell, totalistic cellular automata are only concerned with the sum (or average) of these cells. This reduces the possibility space to a mere 5^{13}, or about 1.2 billion.

3.4. Range

We can also choose the range or neighborhood of the automaton. In the most basic version, cells are only affected by the cells closest to it (and themselves), but we can increase the range so that cells are affected by cells that are further away. Like increasing the number of colors, this dramatically increases the size of the rule space. However, unlike increasing the number of colors, increasing the range does not get us any new output values, so it is of limited usefulness for our purposes. We will keep the range at 1, a "nearest-neighbor" automaton.

3.5. Size of Universe and Boundary Conditions

When calculating or visualizing the propagation of a cellular automaton, it is often treated as if it is expanding into an infinite space. But for musical applications, there are some advantages to restricting the size of the row that the CA is contained within. Each row of a 1-dimensional CA can be used to represent a single measure of music, with each cell representing a point in time within a measure, progressing from left to right (or right to left). For example, a row 16 cells long can be used to represent 16th notes in a measure of 4/4. To change the "time signature," we only need to adjust the length of the row.

The boundary conditions of this universe also have an impact on the patterns created, with a marked difference between fixed boundary conditions (i.e. the edge of the universe always has one value), and periodic boundary conditions (i.e. the edge of the universe wraps around to the opposite side). In my experience, patterns with fixed boundary conditions were more likely to "die out" and result in trivial patterns, while periodic boundary conditions were more likely to create complex patterns or long-term stable patterns more suitable for musical material.

3.6. Finding a Rule

Choosing a rule to use is perhaps the most nebulous part of the process, since there are an overwhelming number of possibilities

to choose from, and not all of those possibilities yield interesting results. Fortunately, because 1-dimensional CA are quick to calculate, software tools can aid in this process immensely, including symbolic computation programs like Mathematica, interactive audio software like Max/MSP, and/or programming languages for visual arts like Processing. Random exploration of CA can often yield surprising and compelling results. Tweaking previously discovered rules is also a useful tactic.

Subjectively, I have noticed that, generally speaking, rules with interesting patterns themselves seem to be more likely to generate interesting patterns. For example, we will use the totalistic 5-color nearest-neighbor rule 0322411423232_5 (also known as code 171592317), which contains some notable features at first glance. There is only one zero, indicating that this is a rare or "special" value in this automaton. The alternation of 2s and 3s suggest that these values will be closely intertwined, and similarly, the symmetrical "4114" construction suggests a relationship between those two values. Again, I am not talking about this automaton in a systematic or rigorous way as a mathematician might want to do, but rather in an intuitive or speculative fashion, in a way that might be useful to a composer or artist.

3.7. Initial Conditions

The initial conditions (or seed) of an automaton can also have a great impact on the automaton's evolution. For example, seeding the automaton described above with random values allows it to evolve into other random-seeding patterns (Figure 3), while seeding it with a a row of the same value allows it to settle into a periodic fluctuation of more rows containing only a single value (Figure 4). Initializing it with the seed 1111211111112111 yields a more intricate and coherent pattern that displays many symmetries and near-symmetries (Figure 5). This seed can be thought of in musical terms as a "backbeat," with accents on beats 2 and 4 of a 4/4 measure. In other words, seeding an automaton with a musically cogent patterns can yield more musically cogent patterns. Another interesting feature of this pattern is that it is a long-term stable structure; after 41 rows, it repeats.

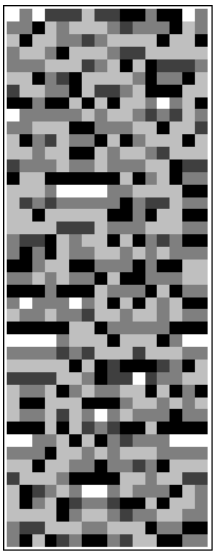


Figure 3. Rule 0322411423232_5 seeded with random values

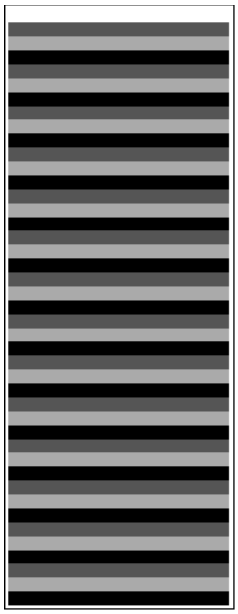


Figure 4. Rule 0322411423232_5 seeded with a single value

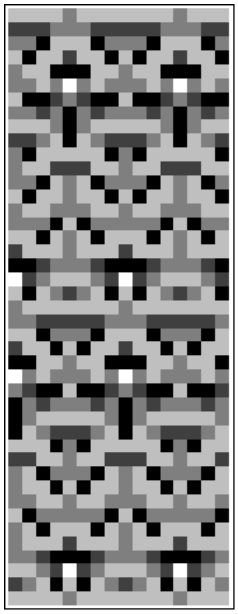


Figure 5. Rule 0322411423232_5 with seed 1111211111112111

4. MUSICAL MAPPINGS

Once we have found a pattern that seems musically viable, the final step is to map this visual information onto sound in a way that is clearly audible.

4.1. Melodic mappings

One intuitive method is to map the different colors to different tones of a scale, with a 5-color automaton mapping onto a pentatonic scale. To prevent the monotony of a constant pulse, tones may be re-articulated only when the next cell changes in value. This creates notes of different duration, and frequently results in syncopated rhythms (Figure 6).

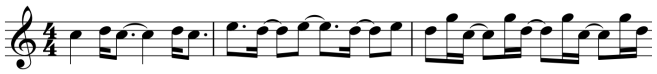


Figure 6. Melodic mapping of first 3 rows of Figure 5

Changing or transposing the scale after the end of a row can create the impression of harmonic changes, allowing for the creation of more large-scale musical structures. These harmonic changes can be applied intuitively by a composer, or by another automaton proceeding at a slower rate.

4.2. Rhythmic mappings

Another method is to use a noise burst or other percussive sample as a sound source, and use automata to manipulate various parameters, including amplitude, resonance, and center frequency of an audio filter. This can create a lively, shifting palette of accents and timbres. These techniques can also be added to melodic mappings in order to enliven the texture. Essentially, any musical parameter that exists on a spectrum can potentially be activated by an automaton.

5. ADVANTAGES OF AUTOMATA FOR GAME AUDIO

All this may seem like a lot of trouble to go to, relative to established methods of scoring and designing sound for games. However, there are a few advantages to using CA for game audio that make them uniquely appealing for certain situations.

5.1. No need for looping audio

One frequent challenge in game audio is to create convincing loops with seamless, unobtrusive endpoints. Unless a scene is tightly scripted, players may take drastically different amounts of time in a certain scene or area, running the risk of looping audio becoming more noticeable the more it repeats. Since CA are dynamically generated and complex, they may continue indefinitely without ever repeating, if desired.

5.2. Transitions are easily accomplished

Similarly, transitions between different musical tracks can be awkward unless carefully synchronized or coordinated. Using CA, transitions between different kinds of musical material can happen at any time simply by changing the rule, the seed, or any aspect of the musical mapping. This transition can happen gradually (through cross-fading or interpolation of values), or instantaneously, even in the middle of a measure.

5.3. Automata are computationally cheap

Since automata can be calculated with simple arithmetic, they add very little to the overall CPU load, an important consideration especially for game engines that use a lot of computational power to render video.

6. LIMITATIONS AND CHALLENGES

Despite these advantages, there are still some limiting factors that need to be overcome eventually for CA to play an effective role in game audio.

6.1. Monotony

Automata-generated music can still have a tendency to become monotonous over longer periods of time. Changes to the rule and the mapping (gradual or immediate) can alleviate this, but human curation and intuition is still a crucial part of this process.

6.2. Style

It seems to be easier to make automata-generated music with certain aesthetic qualities; either ambient, or pulse-based in a minimalist or pseudo-minimalist style. It seems more difficult to use automata to create convincing music in other styles (e.g. jazz). More creative mappings may open up new stylistic territory in the future.

6.3. DSP is not computationally cheap

While automata themselves are computationally cheap, digital signal processing is not. Many game engines and libraries are not particularly optimized for audio, often prioritizing video. This is a problem especially when precise timing is crucial for this kind of musical generation to be convincing. Additionally, game engines do not always give you full control over the signal chain in the manner of audio software.

7. FUTURE WORK

Cellular automata are useful and flexible tools for musical generation that present a great deal of potential for game audio. The rule and mappings described in this article represent a tiny fragment of what is possible. Other potential uses include using automata to synthesize audio, to create sound effects, and to make synesthetic correspondences between audio and visual information. Currently, the main limiting factors are will and imagination.

8. REFERENCES

1. Ariza, C. “Automata Bending: Applications of Dynamic Mutation and Dynamic Rules in Modular One-Dimensional Cellular Automata.” *Computer Music Journal* 31 (2007), no. 1, 29-49.
2. Burraston, D. “Generative Music and Cellular Automata: An Introduction to the Online Bibliography.” *Leonardo* Vol 45 (2), MIT Press, 2012.
3. Fisher, G. “Pixel Paintings.” Mathematical Art Galleries Joint Mathematics Meetings, Atlanta, GA, 2016.
4. Gray, L. “A Mathematician Looks at Wolfram's New Kind of Science.” *Notices of the American Mathematical Society* 50 (2): 200–211, 2013.
5. Hembree, P. J. *Ouroboros and Apocryphal Chrysopoeia: Aesthetics and Techniques.* PhD dissertation, UC San Diego, 2015.
6. Solomos, M. “Cellular automata in Xenakis's music: Theory and Practice,” in *Definitive Proceedings of the International Symposium Iannis Xenakis*. Athens: May 2005-2006.
7. Wolfram, S. “Universality and complexity in cellular automata”, in *Physica D,* volume 10, pp. 1-35, 1984.

Scalable Acceleration of Real-time Audio Processing Using Hardware-Partitioned GPU Compute Units

Carl K. Wakeland
Radeon Technologies Group, AMD.
One AMD Place
Sunnyvale, CA 94085
+1.408.749-2096
carl.wakeland@amd.com

Alexander Lyashevsky
Radeon Technologies Group, AMD.
One AMD Place
Sunnyvale, CA 94085
+1.408.749-4118
alexander.lyashevsky@amd.com

Lakulish Antani
Impulsonic, Inc.
Room 3201, 800 Park Offices Dr.
Research Triangle Park, NC 27709
+1 (650) 434-3198
lakulish@impulsonic.com

ABSTRACT

Creating true presence in virtual reality requires fresh approaches to real-time audio rendering technologies, and these technologies carry significant new demands for real-time computing capacity. New methods such as geometric acoustics require real-time physical modeling, rendered with time-varying convolution – preferably modeling a very high number of sound sources (40 to 60 or more) to reflect the environmental subtlety found in consensus reality. This level of processing can exceed the allowable real-time compute capacity of CPUs found in enthusiast gaming platforms by an order of magnitude or more. Although GPUs possess an immense theoretical computing capacity that is highly capable of enabling these new technologies, efforts to utilize this capacity for scalable, low-latency real-time audio within the context of an intensive game workload have frequently met with unpredictable degradation to graphics performance and other critical functions running in the GPU. AMD TrueAudio Next is a technology that supports scalable, dynamically configurable, hardware-enforced partitioning of compute units (also called shader cores or GPU cores) within single or multiple GPUs, with a dedicated task submission queue. Test results are presented that show how TrueAudio Next can allow intensive low-latency audio workloads such as time-varying convolution with a high channel count to co-exist on a GPU running an intensive virtual reality gaming workload, with predictable performance for both audio and graphics. An audio convolution algorithm from the open-source TrueAudio Next library that provides high GPU utilization will be outlined as an example of a programming method that allows the GPU to process audio with buffer latency as low as 1.3 ms.

Categories and Subject Descriptors

• Computer systems Organization→Architectures→Parallel architectures • Computer systems Organization→Real-time systems→Real-time system architecture • Computer graphics→Graphics systems and interfaces→Graphics processors • Concurrent computing methodologies→Concurrent algorithms.

Keywords

Audio Processing; Game Audio; GPU Compute; GPGPU; Real-time systems

1. INTRODUCTION

Methods for audio processing in gaming for direct and environmental sound effects have by practical necessity been driven primarily by the characteristics of the visual experience, and to a lesser extent by game elements. Until the recent proliferation of virtual reality head-mounted displays, the common visual experience has been defined by flat window displays positioned in front of the user. The constraint of flat window screens as the visual medium for games and entertainment has in effect tempered market demands for fully-immersive and acoustically realistic audio rendering, with a few exceptions found in first-person shooter and horror games. In these cases, accurately positioned audio is not designed specifically for the purpose of creating a sense of presence, but primarily serves tactical needs, such as allowing the player to audibly detect the position of an adversary that is not visible on the screen. To this end, the use of sampled audio, HRTFs, and statistically-modelled, lumped-element acoustic environments has become the most common rendering methodology for gaming audio sound effects.

With the adoption of virtual reality headsets, interest in rendering accurate interactive positional audio has increased, as studies have shown a relationship between use of positional audio in a VR headset and a user's sense of presence [1]. The addition of accurate interactive acoustic modeling to positional audio adds the element of visual/auditory coherence and a higher level of suspension of disbelief. This is gained through auralization of rendered sounds that more closely matches the user's learned expectation of sound auralization in the natural world. Moreover, when accurate auralization is applied to many sound sources in a VR experience, the user is better able to track and distinguish individual sound sources. This enables the creation of more complex and realistic soundscapes in a simulated experience while reducing the risk of creating confusion in the user due to auditory masking effects of competing sound sources.

With the VR market driving new demands for improved audio realism, the question of how to balance this demand with the need for scalable and flexible computing architectures for game audio on personal computers becomes relevant. With mainstream CPU core frequencies having stabilized in the 3 to 5 GHz range, audio developer interest has advanced toward finding methods to reliably scale real-time, low-latency audio processing in the context of multi-threaded, intensive game workloads. A number of studies have been directed at applying parallel compute approaches to the challenge of high-performance audio rendering. For example, Battenberg [2], describes a method of using partitioned convolution algorithms on multi-thread operating systems and multi-core CPUs to support low-latency audio convolution. By necessity, such approaches are dependent on the ability of the operating system and system hardware to provide sufficient real-time performance for deadline-critical processes. This level of reliability is typically straightforward to achieve on a single core, but extensibility to multiple cores may be difficult to achieve when other large workloads, such as a game engine, and

diverse OS features are executing in parallel with real-time audio processing.

A number of researchers have explored the use of the GPU for high-performance audio rendering. For example, Hsu [3] describes methods for using the GPU for real-time, physically modeled finite difference-based sound synthesis. However, the challenge of allowing intensive, low-latency real-time audio workloads on a GPU to be shared with other intensive graphics and compute workloads running in parallel, with deterministic performance for all workloads, has been less explored.

We describe a hardware/software architecture called Compute Unit reservation that enables a configurable subset of compute units (CUs) on a GPU to be dynamically partitioned and reserved for real-time functions such as audio processing, and provide preliminary performance data. We also describe how partitioned convolution can be implemented on a GPU/CPU system and achieve real-time performance.

The rest of the paper is organized as follows. Section 2 provides an overview of GPU compute architecture and prior art for audio use cases. Section 3 provides an overview of Compute Unit Reservation. Section 4 presents an adaptation of overlap-save convolution for low-latency GPU compute implementation. Section 5 presents measured performance results, Section 6 discusses additional optimization strategies, and Section 7 provides concluding remarks.

2. OVERVIEW OF GPU COMPUTE ARCHITECTURE AND AUDIO USE CASE

2.1 Conceptual Overview

GPU Compute is the application of highly parallel processing technology, originally developed for graphics rendering, to general purpose computing applications. GPU compute architectures favor very wide (typically 64- or 32-lane) vector SIMD instruction execution over superscalar approaches such as predictive branching and speculative execution. These choices provide GPU compute architectures with higher maximum throughput than contemporaneous superscalar architectures. Signal processing workloads that intrinsically support a high degree of data parallelism are best equipped to approach the maximum throughput ratings of GPU compute architectures.

As described in [4], GPU Compute architectures conventionally schedule and execute kernel workloads under hierarchical partitions called workgroups (also called “blocks”), which are composed of wavefronts (also called “warps”), which in turn are composed of work items (also called “threads”). During GPU program kernel execution, a scheduler creates workgroups and assigns them to CUs. Wavefronts from these workgroups are scheduled on their assigned CUs as they become ready for execution. More than one workgroup may share a CU, if necessary. CUs are provided with register banks, local shared data, and L1 cache. Multiple CUs may share an L2 cache to system memory in addition to GPU global shared memory.

Highly data parallel tasks can map to multiple workgroups and execute on all CUs on a GPU device in parallel. When this highly parallel mapping occurs, tasks are able to execute in the minimum cycle count possible for the GPU device. For graphics shading tasks that typically work on tiles that are subdivided from a larger image, achieving parallelization across all CUs is very common. Due to limited parallelism in some algorithms, serial dependencies, and the phenomenon called divergence, in which some work items in a wavefront may be idle, it is possible for full utilization of all CUs to not be achieved by a kernel at a given time. Some GPU architectures, such as the AMD Graphics Core Next (GCN) architecture, allow wavefronts on CUs from the scheduler to be swapped out if their parent queue has exceeded its time quanta (due to the size of the job, serial dependencies, or divergence within the waves) or is being replaced by wavefronts from an arriving higher priority queue. This capability is called Asynchronous Compute with instruction-based preemption.

2.2 Audio Use Cases on GPU

In the GCN architecture, when a workgroup kernel or graphics shader is running on all CUs and is not swapped out as described above, it continues to occupy all CUs until kernel execution is complete. In the environment of graphics shaders, the size and execution time of shader kernels is typically limited to the size of draw calls that are supported by the graphics rendering API in use. The execution time of GPU compute kernels is typically bound by programmatic limitations and compiler conventions that avoid the creation of persistent kernels that could have unbounded execution time. For workloads such as graphics processing that are more throughput-critical than latency-critical, these built-in safeguards are able to maintain excellent overall system efficiency. Workloads that are highly latency-critical, such as audio rendering that runs over a fixed time quantum, may be at a disadvantage, however, if any kernel execution time exceeds the audio time quantum period. For example, a gaming audio streaming buffer must typically be processed within a strictly defined time slice taken from periodic time quanta of 20.8 ms or 10.4 ms. If any competing kernel running concurrently with such an audio process requires more than the audio time quanta period to complete on the entire GPU, a user-perceptible audio dropout or sound glitch occurs.

Through an extension of Asynchronous Compute capability, AMD GCN GPU schedulers allow multiple software-visible task queues to be supported, with different levels of priority assigned to these queues during software configuration. This approach can minimize queue waiting time and improve latency for deadline-constrained kernels, but cannot avoid latency overhead that results from kernels with prolonged execution times that utilize all of the CUs on a GPU.

In physics-based audio rendering approaches such as geometric acoustics [5], there can be a mixture of processes with varying deadline dependencies, some that are considered as critical “hard” real-time, such as convolution performed on streaming audio samples, and some that are less-critical “soft” real-time. The latter can include audio propagation calculations that result in periodic updates to time-varying impulse response (IR) kernels that are referenced by per-sound-source streaming convolution that is performed in hard real-time. Since a missed update to a time-varying convolution kernel does not result in an audio dropout or glitch, but rather a much less perceptible delay to the update of an audio filter response, it is possible to double-buffer these IR kernel updates for purposes of latency hiding. By contrast, the aforementioned hard real-time streaming audio buffers cannot be afforded the same benefit of latency-hiding, but must be processed in the same time quanta that they arrive in.

At a finer granularity, different subcomponents of spatial audio simulation can have different latency requirements. At one extreme, direct sound and direct path occlusion must be simulated with low latency, since the user may see sound sources move at

very high visual frame rates, and any obvious mismatch between the visuals and the audio will lead to a significant reduction in presence. At the other extreme, broad-scale environmental effects like reverberation – whether modeled manually using reverb volumes, or using physically-based approaches with coarse spatial sampling – can tolerate relatively high latencies[6], since these effects vary more smoothly as the user moves through the virtual environment. With head-mounted displays, the "rotational latency" – the latency experienced by the user between turning their head, and hearing the sound field respond to match – becomes increasingly important. However, Ambisonics offers one solution to this challenge: instead of repeating simulation or convolution based on the user's head rotation, we can just apply spherical harmonics rotation algorithms to the final Ambisonics mix, allowing the rotation-related calculations to run at the very end of the audio processing pipeline and independent of the number of sources or the geometric complexity of the environment, thus significantly reducing latency.

Ambisonics [7] is increasingly becoming the format of choice for VR and 360 video. As a format for encoding spatial audio, it offers several significant advantages: a) the level of spatial or directional detail can be scaled as per the performance characteristics of the user's hardware platform – this is accomplished by decoding up to an appropriate maximum *order* of Ambisonics; b) it can be efficiently converted to either multichannel surround streams, or headphone-based binaural streams for rendering; and c) the sound field described by an Ambisonics stream can be efficiently rotated to match the user's orientation in the virtual and/or physical environment. As Ambisonics finds increasing use in simulating spatial and environmental audio (for example, by using Ambisonics-encoded impulse responses to encode environmental reflections and reverb), the computational requirements of IR convolution increase quadratically with the Ambisonics order. This makes it especially challenging to meet the tight time constraints of IR convolution, without using highly data-parallel devices like GPUs.

3. COMPUTE UNIT RESERVATION

Compute Unit reservation (CU reservation) is a topological modification to the heretofore homogeneous GPU CU array. The architectural concept of CU reservation is to enable a limited portion of a GPU's CU array to be exclusively allocated to a specific task queue, called a real-time queue (RTQ). The CU allocation and RTQ may be reserved and released dynamically at runtime, thus allowing specific applications to opt-in or out. Task mapping is one-to-one between the RTQ and the block of reserved CUs, meaning that tasks submitted to the RTQ are mapped to workgroups that are executed solely on the reserved CUs, and tasks submitted to other queues cannot use the reserved CUs. All tasks may share global memory and L2 cache.

3.1 Example topology

Topologies for CU reservation are scalable with the sizes of GPUs. An example implementation topology of CU reservation is shown in Figure 1. In the figure, four CUs from an array of 40 in the CU array on the right side are reserved and allocated exclusively to dedicated hardware queue slots for real-time execution. Software writes all tasks intended to execute on the block of reserved CUs to the RTQ, shown at top left. The scheduler creates workgroups for tasks from the RTQ and dispatches them to the block of reserved CUs. All other tasks and queues use the remainder of the CU array, but can use queues with different priority levels. Global memory resources such as L2 and global shared memory continue to be shared by all CUs in the array. Thus there is a nonfinite risk of contention for memory resources between the reserved CUs and the remainder of the array. Our results to date (section 5) have not shown the impact of this contention to be significant for convolution workloads.

The number of CUs that are reserved may vary on different GPUs. Typically for a gaming type of discrete GPU we consider reserving 4 to 8 CUs for critical real-time audio rendering tasks that directly affect audio sample buffers, such as convolution rendering.

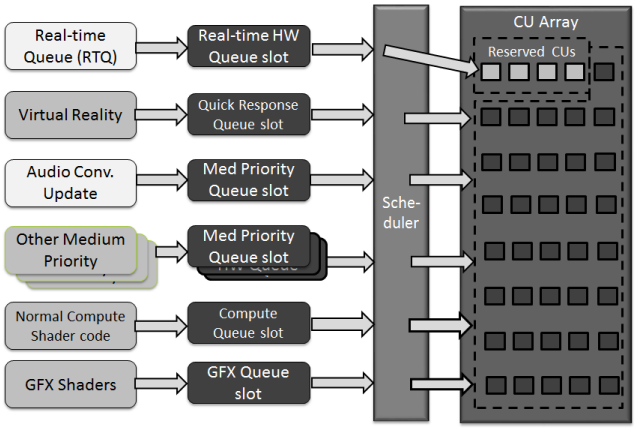


Figure 1: Compute Unit Reservation within Compute Unit Array

In Figure 1, we see that of two highlighted audio tasks seen in the leftmost column, the audio convolution update task is directed into a "medium priority" queue, which shares the main pool of CUs with all other compute and graphics shader tasks. Using this queue gives the convolution IR update tasks (which reflect the results of audio propagation computations) very good quality of service to ensure that filter updates reflect the movement of sound sources and listener position in a game audio scenario. At the same time, higher priority virtual reality display position updates are not impeded by this workload. The RTQ receives the critical real-time tasks that implement convolution rendering for each audio stream; using the reserved pool of CUs ensures that these tasks are not interrupted, preventing audio dropouts.

In the case of Geometric Acoustics, the propagation calculations (e.g., ray tracing, specular reflections, acoustic radiance functions, etc.) would be mapped to the medium priority or general compute queues, which receive the same priority as the RTQ. The resulting impulse responses are sent to the low-latency per-stream convolution functions which run on the RTQ.

4. LOW-LATENCY CONVOLUTION ALGORITHM FOR GPU

Here we describe the low-latency convolution algorithm that we

GameSoundCon'16, Sept 27-28, 2016, Los Angeles, CA, USA.

adapted for optimal performance on GPU for audio use cases.

4.1 Single-channel algorithm

We start with the description of a single channel algorithm. A multiple-stream system and overall performance considerations are discussed later in the section.

Our algorithm assumes the length of the input/output buffer (audio frame) is held constant for the duration of processing and is a power of 2. It assumes also the length of the IR is fixed and is multiple of the frame length.

The algorithm utilizes an invertible transformation T obeying the convolution property:

FT(x © y) = FT(x) * FT(y)

This is equivalent to:

(x © y) = IT(FT(x) * FT(y)),

Where © is a convolution, * is an element-wise multiplication in the transformed space, FT is a forward transform, and IT is the inverse transform of T.

Both the FFT and FHT (Fast Hartley Transform) conform to the property and can be utilized. Specific considerations of transform T will be discussed in section 4.2.

The algorithm combines the overlap-save and the uniformed partitioning [2] convolutional methods to reduce latency and computational complexity.

The algorithm is comprised of two computationally independent parts: an impulse response (IR) update and audio frame processing:

4.1.1 IR Update

a) IR data is divided into blocks. Each block is the size of the audio frame. A block is padded with 0s to double the block size.
b) The block is transformed into the frequency domain with the transform T.
c) All blocks are concatenated to form the IR frequency domain buffer – IRFDB.

4.1.2 Audio frame processing

Refer to Figure 2.

a) An incoming audio frame is concatenated with the frame data from a previous round, forming a double-size buffer. If it is the first frame, it is zero-padded.
b) The buffer is transformed into the frequency domain with the transform T.
c) The frequency data block is written into a cyclic "data frequency domain history buffer" – DFDHB, as the newest buffer, with the oldest buffer being overwritten. The number of blocks in DFDHB is equal to the number of blocks in IRFDB.
d) A block-by-block element-wise complex multiply is performed between data blocks from DFDHB and IRFDB.
e) Blocks from the IRFDB are selected in time-forward order from the start to the end. Blocks from the DFDHB are selected in time-reverse order, starting from the newest block and proceeding in backward sequence to the oldest block.
f) The resulting blocks are summed up to produce a single block with a size of 2 audio frames.
g) This block is inversely transformed into the time domain with the transform T.
h) The second part of the new time-domain block is discarded and the first part is copied into the output buffer.
i) The output buffer now contains the convolved data.

4.1.3 System considerations

The algorithms described above can be scaled into any number of streams, constrained only by the latency requirement.

There are few important parameters defining algorithm behavior and complexity at system level:

a) The maximum number of channels that can be processed at once, which also defines the maximum number of IR filters,
b) The length of an audio frame,
c) The maximum length of the IR.

The system has been designed to handle the IR update and the audio frame processing independently and asynchronously.

The IR update may take much more time than the strictly time-limited audio frame processing. To avoid stalling the audio frame processing due to IR update the system employs a double

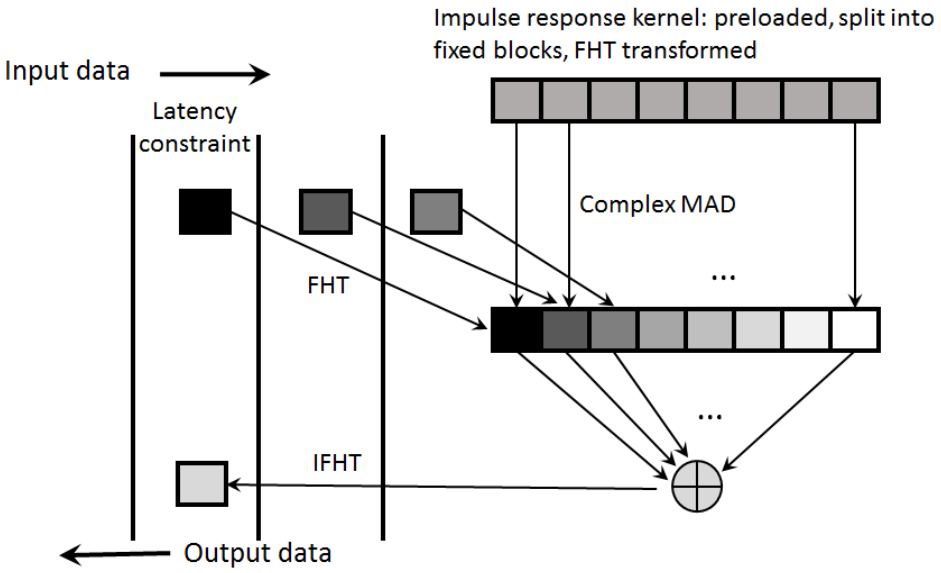


Figure 2: GPU Uniform Partitioned Convolution

buffering scheme, with the IR buffer ID being assigned to each IR. The IR's buffer ID is a position of the IR in IRFDB. The IR update algorithm is supplied with the control map of currently "active" IRs with their ID. The map can be changed at every IR update cycle.

The audio processing algorithm on GPU is also supplied with its own control map that assigns the IR's ID to the audio stream the IR is going to be convolved with. The map can be changed at each processing round.

4.2 Computational complexity, latency and throughput

First, we estimate the theoretical complexity of the algorithm and consider the selection of the basic transform. Second, we compare the properties of the IR update and audio processing portions, and thirdly we explore the optimal mapping of the two portions of the algorithm onto the GPU topology.

If the number of audio streams being processed is denoted by "S', then in a typical deployment of stereo audio streams, the algorithm is going to process S*2 = C channels of audio. Both the FHT and FFT can be used as a basis for the algorithm. Their computational complexity is proportional to $(N/2)\log_2(N)$, where N is the number of samples in one time-domain audio frame. Essentially it is the number of loops in a classical "butterfly"-type transform algorithm. To get a more precise estimate we need to scale it by the number of arithmetic operations per loop.

The implementation of both FFT and FHT combines multiplications with twiddle factors, summation and address calculation and requires a minimum of 12 FLOPS for FHT and 10 for FFT.

The FFT appears to be more optimal, but the transforms (both direct and inverse) are not the only parts of the convolution algorithm, and to make the overall best choice we have to consider other aspects of the full algorithm and the implications of its mapping on GPU. Assuming C channels with maximum size of IR = K * N, we have the following:

The audio frame processing algorithm consists of 3 parts:

1. Direct transform of C blocks of N*2 real numbers;
2. Complex multiply and add of C*K*N*2 IRFDBs and DFDHB;
3. Inverse transform of C blocks of N*2 reals.

The IR update is a direct transform of C*K blocks of N*2 reals. The FHT keeps data inside the R(real) space in both the time and frequency domains.

For us it has 3 advantages

a) 2X local memory footprint reduction,
b) 2X memory bandwidth reduction,
c) 2X global (video) memory footprint reduction.

Advantage (a) allows us to keep all intermediate data for both forward and inverse transforms in the local, CU-attached GPU memory, significantly improving the performance of the transforms. Both IR update and parts 1 and 3 of the audio processing have taken advantage of this property.

If C is large, the audio processing algorithm is completely dominated by the 2nd part and advantage (b) reduces the overall time of the audio processing almost 2-fold.

Although advantage (c) does not have a direct performance implication, it drastically reduces the overall graphics memory requirement that is comprised of a double-buffered IRFDB and a single DFDHB with 3*C*K*N*4 total number of bytes.

Considering all the above advantages, we've selected FHT as our basis transform in spite of a small increase in computational complexity.

As a backup and to be compatible with popular GPU packages we have an FFT implementation of all parts of the convolution algorithm utilizing the AMD clFFT library.

Based on the convolution algorithm described above we can estimate the theoretical computational complexity of the entire convolution process. N is an audio frame size:

a) IR update complexity: $12*N*\log_2(N*2) * C * K *$IR update frequency.(in Hertz)
b) Audio processing complexity: $(12*N*\log_2(N*2) * 2 + 10 * K * N * 2) * C *$ Frame frequency (in Hertz).

The first term in summation (b) is the computational complexity of forward and inverse transforms.

The second term in the summation (b) is the multiply and add complexity for the frequency-domain convolution with FHT.

Memory transfer required for the audio processing convolution step can be estimated as:

Memory transferred (bytes) = (N*2*2 + 2*K*N*2 + K*N*2/32) *C *4

In this formula, the first term covers the size of the input buffer (read and write), the second term covers the data that must be accessed to perform the complex multiply-add between the filter and the history buffer of the convolution, and the third term is the write size of the resulting data.

It is typically found that audio convolution is memory-bound on the current generation of GPUs. This may change with future GPUs as new generations of memory and bus technologies continue to improve bandwidth. For the present time, when other steps such as the aforementioned audio propagation calculations are also performed on the GPU, the amount of data that has to transfer on external busses such as PCIe is minimized, with potential gains in overall throughput.

4.3 GPU Mapping

With the introduction of the RTQ a single GPU device can be viewed as a runtime-reconfigurable multi-node computational device.

The IR update and audio processing steps are independent processes and can be run in parallel on different nodes with the new GPU topology. Since the IR update is not a strictly deadline-critical process it can run on the normal compute or on a medium-priority queue while the audio processing runs on RTQ.

While running on the normal compute queue, however, the IR update may adversely impact the graphical throughput. It can be advantageous to run both the audio processing and IR update portions on the RTQ if the there is enough throughput available. This approach keeps the audio processing workload largely a separate effort to develop from the graphics workload, which can simplify the logistics of having the graphics and audio components of a game design project proceed independently. The graphics workload sees a reduction in shader compute capacity due to some CUs becoming unavailable, but the impact is deterministic since the reservation size can be established early in the development process to avoid unintended increases in audio

usage that could affect graphics performance. If, due to incremental changes in the audio design, the audio workload exceeds the previously agreed-to CU compute capacity needed to achieve real-time, only the audio workload is affected by the deficit – the graphics CUs are not allowed to be used to increase the audio compute capacity beyond the fixed allotment. The audio designer can resolve the deficit by either reducing the audio workload or negotiating an increased budget of CUs for audio with the graphics developers. Thus the overall system achieves a higher level of deterministic execution.

The following 2 inequalities should hold to allow both portions of the audio convolution algorithm to be executed on the RTQ:

1. Algorithm time = (IR update time (on 1 CU) * IR update frequency + audio frame convolution time(on 1 CU) * frame frequency) / number reserved CUs << 1s

2. Maximum latency = (IR update time (on 1 CU) + convolution time(on 1 CU)) / number reserved CUs << 1/ frame frequency

We assume a linear scaling with respect to number of CUs. This assumption is very well supported by the nature of the algorithm and by statistics we've gathered so far.

Both parts of the algorithm run on the same set of CUs and thus have to run serially relative to each other.

The 1st inequality reflects the fact that the total algorithmic complexity should not exceed the total throughput of dedicated CUs.

The 2nd inequality guarantees that the IR update and the audio frame processing running back to back can be competed in less than 1 audio real-time interval.

The 2nd condition is much more restrictive than the 1st and could limit the number of streams the 1st inequality would allow.

5. RESULTS

To test our GPU-based convolution use case with CU reservation, we used an AMD Radeon™ R9 Fury X GPU. We tested 4 CU, 8 CU and all-CU configurations, with 64 stereo audio streams at 48 kHz, single-precision floating-point samples, with 1024-sample buffers. The GPU clock frequency for the CU array was 1050 MHz. Using the open-source CodeXL profiler, we measured the average execution times of an OpenCL kernel that computed the audio processing portion of our GPU-based convolution. The results are shown in Figure 3 and Figure 4. We can remark that the execution times are seen to scale linearly with the number of CUs, with essentially a halving of execution times for 8 CUs vs. 4 CUs. Note that for real-time implementation, the time quantum for 1024 samples at 48 kHz is 21.3 ms. The time length of each of the four convolution kernels is computed as 1/48000 * IR_length; thus for a 64K kernel, the time length is 65536/48000 = 1.36 seconds, and for a 512K length kernel, it is 10.92 s.

To test the effectiveness of CU reservation to minimize variation in kernel execution times, we compared the ratio of worst-case to average kernel execution times when running convolution instances on 4 CU and 8 CU RTQs, and on the normal all-CU queue. The results are shown in Figure 5, Figure 6 and Figure 7. It is observed that the ratios are close to 1 when running on the RTQ, but significantly higher on the normal CU array, and the ratio increases when a significant graphics workload is competing with the audio workload running on the general CU array. It can be remarked that the RTQ showed significantly higher execution time consistency; this can be attributed to the isolation of the audio workload on the RTQ.

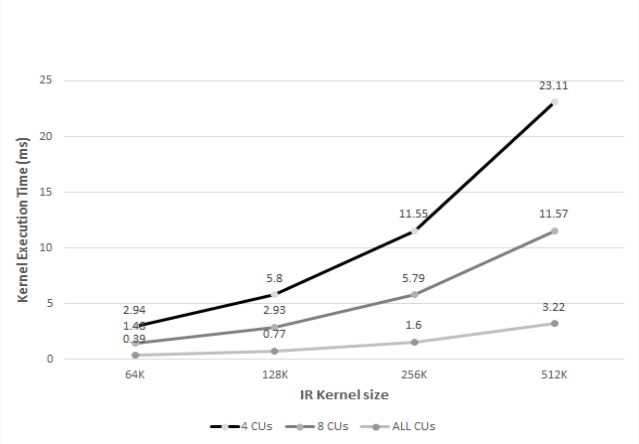


Figure 3: 128-Channel Convolution Processing Execution Times, Varying IR Kernel Sizes

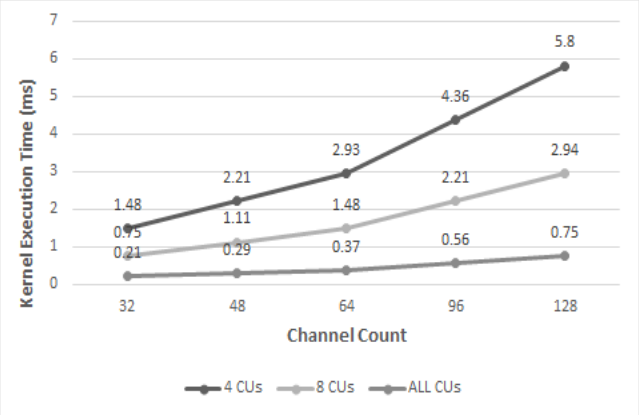


Figure 4: 128K-sample IR Convolution Processing Execution Times, Varying Channel Count

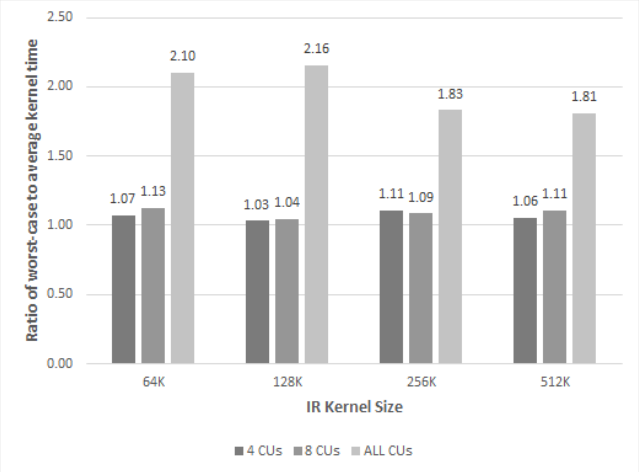


Figure 5: Ratio of worst-case to average kernel execution time, 128 channels, varying kernel size, minimal competing workload

We also measured execution times for IR updates. For a 64K kernel, the update times were 19.16 ms for 4 CUs, 9.55 ms for 8 CUs, and 1.26 ms for the complete CU Array.

6. ADDITIONAL OPTIMIZATION STRATEGIES

In this section, we provide recommendations for managing IR updates with audio convolution processing on a limited set of reserved CUs.

6.1 Non-uniform convolution for high frequency audio samples.

Lower latency audio processing uses smaller audio frames. The uniform convolutional method we've employed does not scale linearly with the size of the audio frames. This outcome is related to the fact that the multiply and add operation dominates the audio

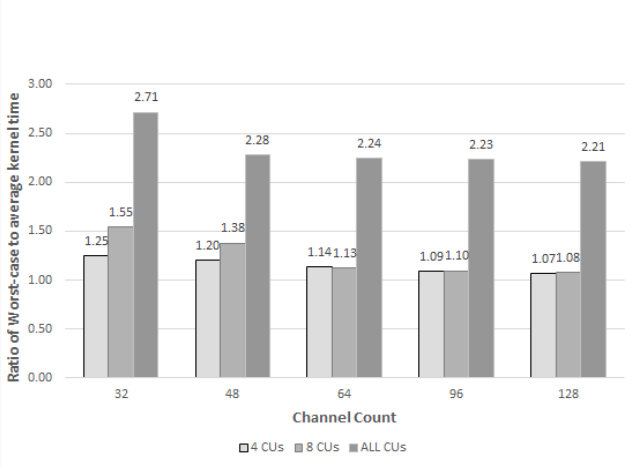


Figure 6: Ratio of worst-case to average kernel execution time, 128K sample IR, varying channel count, minimal competing workload

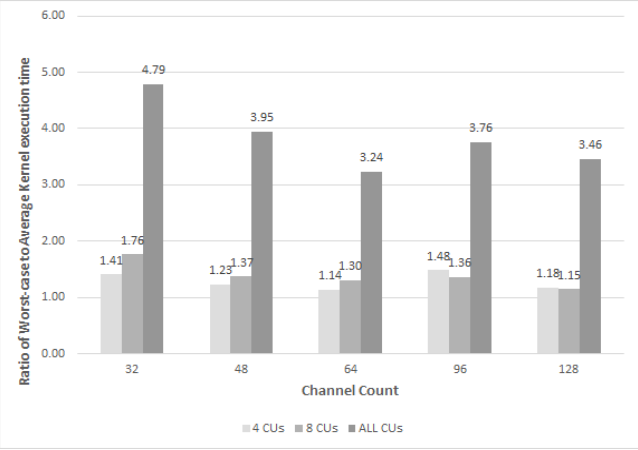


Figure 7: Ratio of worst-case to average kernel execution time, 128K sample IR, varying channel count, significant competing workload

processing and is linearly proportional to the size of the filter and almost independent of the audio frame size. It is possible to render very low latency audio (e.g., 1.3 ms buffers) with this scheme, but GPU rendering becomes increasingly less efficient as the buffer length decreases below 5 to 10 ms. To improve efficiency for smaller audio frames down to 1.3 ms, we can use the non-uniform convolution method. However, instead of using several cores as described in [2], we can exploit the fact that direct/inverse transforms are extremely inexpensive on GPU relative to the costly memory-bound complex multiply and add. For small audio frames, we can consider running 2 partial convolutions of different sizes on the same device and to reduce the latency of the multiply and add about 10 times.

6.2 Distributing the IR update over multiple audio processing rounds.

The 2nd inequality described in sub-section 4.3 constrains the ability to achieve glitch-free low latency audio when both the IR updates of a large convolution filter and latency-critical audio processing are targeting the same set of reserved CUs, although the total throughput of the CUs might allow it. To improve the audio processing latency we suggest to split the IR update into a few stages each of which satisfies the 2nd inequality. This can be viewed as a programmed preemption of the IR update function by the higher-priority audio processing function. The IR update control map described in section 4.allows this distribution to be easily realized, at the cost of increasing the IR update latency.

7. CONCLUDING REMARKS

Our results show that low-latency audio convolution of a high number of audio streams is within reach of contemporary GPUs. Moreover, through the use of CU reservation, the execution time variance of audio convolution can be reduced. The latter is a highly valuable prerequisite for achieving practical use of this capability when it is deployed alongside intensive graphics workloads. Utilization of this capability can be especially beneficial to convolution rendering used in physically-based audio rendering algorithms such as geometric acoustics.

8. ACKNOWLEDGEMENTS

The authors would like to thank Gabor Sines, Geoffrey Park, Michael Mantor, Anish Chandak, AMD, and Impulsonic, Inc. for their review and support.

9. REFERENCES

[1] P. Larsson, A. Väljamäe, D. Västfjäll, and M. Kleiner. 2005. Auditory-induced presence in mediated environments and related technology. In *Handbook of Presence*. Lawrence Erlbaum.

[2] Eric Battenberg, Rimas Avizienis. 2011 Implementing Real-time Partitioned Convolution Algorithms on Conventional Operating Systems. *Proc. of the 14th Int. Conference on Digital Audio Effects* (Paris, France, Sept. 19-23, 2011).

[3] Bill Hsu and Marc Sosnick-Perez. 2013 Finite difference-based sound synthesis using graphics processors. In *ACM Queue Vol 11, Issue 4*, May 8 2013. http://queue.acm.org/detail.cfm?id=2484010

[4] Kyoshin Choo, William Panlener, and Byunghyun Jang. Understanding and Optimizing GPU Cache Memory Performance for Compute Workloads. In *ISPDC '14 Proceedings of the 2014 IEEE 13th International Symposium on Parallel and Distributed Computing* DOI=http://doi.acm.org/10.1109/ISPDC.2014.29

[5] Lakulish Antani, Anish Chandak, Lauri Savioja, and Dinesh Manocha. 2012. Interactive Sound Propagation using Compact Acoustic Transfer Operators. *ACM Transactions on Graphics.* Volume 31 Issue 1 DOI=http://doi.acm.org/10.1145/2077341.2077348

[6] Interactive 3D Audio Rendering Guidelines, Level 2.0. http://www.iasig.net/pubs/3dl2v1a.pdf

[7] AmbiX: A Suggested Ambisonics Format http://iem.kug.ac.at/fileadmin/media/iem/projects/2011/ambisonics11_nachbar_zotter_sontacchi_deleflie.pdf

SPEAKER BIOS

Simon Calle is a musician, sound designer and audio engineer for Explode Studio in Brooklyn, NY. He is currently an active member of the New York University Immersive Audio Interest Group, led by Dr. Agnieszka Roginska. Together, the team is producing immersive audio and visual experiences using binaural and ambisonic recording techniques. As a researcher, his main focus lies in the development of tools for the integration of 3D procedural audio processing with VR games. He is also conducting research into implementing 3D audio techniques for VR and AR with future sound design workflows. He recently got named the Emil Torick Scholar for 2016 by the Audio Engineering Society.

Paul Hembree is active as a composer, creative music technologist, and educator. His recent music includes Cerebral Hyphomycosis (2016), a duo for real and virtual cellists, and Ikarus- Azur (2013), a La Jolla Symphony and Chorus commission. His audiovisual improvisations were featured at National Sawdust on the 2016 New York City Electroacoustic Music Festival, as part of the New York Philharmonic's biennial celebrations. He is a sought after collaborator, and is currently engaged as sound designer for Pulitzer prize winning composer Roger Reynolds' FLiGHT project, working alongside the JACK Quartet and video artist Ross Karre. In 2015 he received his PhD in music, specializing in composition and computer music, from the University of California, San Diego, where he also taught music theory, history, composition, and game audio courses.

Sam Hughes is a sound designer and voice actor who has worked on various projects in both disciplines for over many years. After completing an Undergraduate degree of BSc (Hons) Music Technology and Audio Systems with a 2:1 at The University of Huddersfield, Sam freelanced on feature films and short films as a sound designer whilst running in London. In May 2013 , Sam founded award nominated site, The Sound Architect™, with the goal of meeting other audio professionals and sharing their knowledge and experiences with the rest of the audio community. In the same year Sam was selected as one of the first to ever receive the Prince William Scholarship from both BAFTA & Warner Bros. to study MSc Post Production with Post Production at The University of York. Sam graduated with a Distinction and a Departmental Award for achieving the Highest Overall Average on the course. Sam has presented research from this degree at the AES 56th Annual Audio for Games Conference. Sam has always been actively involved in the audio community, writing for BAFTA Guru, sitting on the BAFTA Youth Board, being a member of BAFTA Crew Games, organising industry events such as the monthly meet up in the North of England called Game Audio North or GAN, sitting on the AES Audio for Games Advisory Board, and being a core member of the site DesigningMusicNow. Sam is currently taking his audio skills and interest further by embarking on a fully funded PhD scholarship with Intelligent Games and Games Intellegince (IGGI) to study spatial audio and its effects on player emotion, titled "Affect and Emotion Using Immersive Sound Design in Intelligent Games" at the University of York.

Sally-anne Kellaway is the Senior Audio Designer for Zero Latency, a Melbourne-based VR developer that hosts free-roam room scale VR games for 6+ players, with additional sites opening in Japan and in unannounced locations worldwide. She also works closely with the Industry-standard Audio Middleware solution FMOD, is the Sound Supervisor for the 360 Uku VR Documentary, and is a seasoned speaker at major Game Development conferences worldwide. Sally has worked for SEGA Studios Australia and has completed a Masters in Design Science with a specialisation in Audio and Acoustics, marrying industry and academic knowledge of Psychoactoustics to bring premium immersive audio technology to VR.

Bill Kapralos is an Associate Professor in (and former Program Director of) the Game Development and Entrepreneurship Program at the University of Ontario Institute of Technology. His current research interests include: serious games, multi-modal virtual environments/reality, the perception of auditory events, and 3D (spatial) sound generation for interactive virtual environments and serious games. He has led several large interdisciplinary and international serious gaming research projects that have included experts from medicine/surgery, and medical education with funding from a variety of government and industry sources. He is currently leading the serious gaming theme within the Social Sciences and Humanities Research Council of Canada (SSHRC) Interactive and Multi-Modal Experience Research Syndicate (IMMERSe) initiative.

Isaac Schankler is a composer and music researcher who has created music for a variety of independent games, including Analogue: A Hate Story, Depression Quest, and Redshirt. He is Assistant Professor of Music at Cal Poly Pomona, and Artistic Director of the concert series People Inside Electronics. Previously, he has presented talks on games and music at IndieCade East, Different Games, and the New Music Gathering.

Carl Wakeland is an AMD Fellow, currently leading the Perceptual Computing Architecture Group within AMD's Radeon Technologies Group. Carl has been involved with audio processing for gaming for 19 years. Carl is the audio domain architect and strategist at AMD, where he has led architecture for both console and PC audio accelerators and hardware. His background includes 10 years at Creative Labs/Emu Systems, where he was a lead DSP architect, designer, and ASIC design director on Creative Labs SoundBlaster ASICs, working on successive game sound accelerator generations from EMU10K1 through EMU20K2, as well as the SoundCore 3D. Carl was the audio architect and sound designer for AMD's innovative Surround House 360 experiences at CES, and is now focused on acceleration of leading-edge audio algorithms for virtual reality. Carl holds 28 US patents, and has a degree in Electrical Engineering and a degree in Music Performance from the University of Hawaii. Carl has presented at GDC VRDC and at multiple AMD Developer Summits.