D^4: an Interactive 3D Audio Rapid Prototyping and Transportable Rendering Environment Using High Density Loudspeaker Arrays

Ivica Ico Bukvic
Virginia Tech
SOPA, DISIS, ICAT
ico@vt.edu

ABSTRACT

With a growing number of multimedia venues and research spaces equipped with High Density Loudspeaker Arrays, there is a need for an integrative 3D audio spatialization system that offers both a scalable spatialization algorithm and a battery of supporting rapid prototyping tools for time-based editing, rendering, and interactive low-latency manipulation. D^4 library aims to assist this newfound whitespace by introducing a Layer Based Amplitude Panning algorithm and a collection of rapid prototyping tools for the 3D time-based audio spatialization and data sonification. The ensuing ecosystem is designed to be transportable and scalable. It supports a broad array of configurations, from monophonic to High Density Loudspeaker Arrays. D^4's rapid prototyping tools leverage oculocentric strategies to importing and spatially rendering multidimensional data and offer an array of new approaches to time-based spatial parameter editing and representation. The following paper focuses primarily on the unique affordances of its rapid prototyping tools.

1. INTRODUCTION

The history of Western music can be seen as a series of milestones by which the human society has emancipated various dimensions of aural perception. Starting with pitch and rhythm as fundamental dimensions, and moving onto their derivatives, such as homophony, and polyphony, throughout the centuries each new dimension was introduced and refined until its level of importance matched that of other already emancipated dimensions. In this paper the author posits that the observed maturity or the emancipation of these dimensions is reflected in their ability to carry structural importance within a musical composition, such as a pitch set becoming a motive, or a phrase that is further developed and varied and whose permutations can independently drive the structural development. The same structural importance can be also translated into research contexts where, for instance, a significant component of the data sonification can be conveyed within the emancipated dimension. With the aforesaid definition in mind, even though timbre plays an important role in the development of the Western music, particularly the orchestra, its steady use as a structural element does not occur until the 20th century. Indeed, the 20th century can be seen as the emancipation of timbre. Similarly, while the audio spatialization has played a role throughout the history of music, with occassional spikes in its importance, including the Venetian cori spezzati or the spatial interplay among the orchestral choirs, its structural utilization is a relatively recent phenomena. Today, the last remaining dimension of the human aural perception yet to undergo its emancipation is spatialization. From augmented (AR) and virtual reality (VR), and other on- and in-ear implementations, to a growing number of venues supporting High Density Loudspeaker Arrays (HDLAs), 21st century is poised to bring the same kind of emancipation to the spatialization as the 20th century did to timbre. Similarly,sSpatialization is yet to attain the same kind of emancipation in the area of data experience using audification and sonification [1]. Both are relatively new but nonetheless thriving research areas whose full potential is yet to be realized.

In this paper HDLAs are defined as loudspeaker configurations of 24+ loudspeakers capable of rendering 3D sound without having to rely solely on virtual sources or post-processing techniques. This definition suggests multiple layers of loudspeakers spread around the listening area's perimeter.

Apart from the ubiquitous amplitude panning [2], contemporary audio spatialization algorithms include Ambisonics [3], Head Related Transfer Function (HRTF) [4], Vector Based Amplitude Panning (VBAP) [5], Depth Based Amplitude Panning (DBAP) [6], Manifold-Interface Amplitude Panning (MIAP) [7], and Wave Field Synthesis (WFS) [8]. The following paper focuses primarily on spatialization strategies that are reproducible in physical environments and leverage venue's physical affordances with minimal amount of idiosyncrasies, such as the vantage point without requiring additional technological support, e.g. a motion tracking system.

There is a growing number of tools that leverage the aforesaid algorithms. This is of particular interest because the lack of such tools makes it particularly cumbersome to integrate algorithms in the well-established research and artistic production pipelines. The most common implementations are found in programming languages like Max [9] and Pure-Data [10] where they offer spatialization capabilities (e.g. azimuth and elevation), leaving it up to user to provide more advanced time-based editing and playback. Others focus on plugins for digital audio workstations (DAW) [11, 12] leveraging associated tool's automation, or offer a self-standing applications dedicated to audio rendering, such as Sound Particles [13], Meyer's Cuestation [14], Zirkonium [15], and Sound Emotion's Wave 1 [16]. The fact that a majority of these tools have been developed in the past decade points to a rapidly developing field. A review of the existing tools has uncovered a whitespace [17], a unique set of desirable features an algorithm coupled with time-editing tools ought to deliver in order to foster a mode widespread adoption and with it standardization:

- The support for irregular High Density Loudspeaker Arrays;
- Focus on the ground truth with minimal amount of idiosyncrasies;
- Leverage vantage point to promote data comprehension;
- Optimized, lean, scalable, and accessible, and
- Ease of use through supporting rapid-prototyping time-based tools.

2. D^4

D^4 is a new Max [9] spatialization library that aims to address the aforesaid whitespace by:

1. Introducing a new lean, transportable, and scalable Layer Based Amplitude Panning (LBAP) audio spatialization algorithm capable of scaling from monophonic to HLDA environments, with particular focus on advanced perimeter-based spatial manipulations of sound that may prove particularly useful in audification and sonification scenarios, and
2. Providing a collection of supporting rapid prototyping time-based tools that leverage the newfound audio spatialization algorithm and enable users to efficiently design and deploy complex spatial audio images.

D^4's Layer Based Amplitude Panning (LBAP) algorithm groups speakers according to their horizontal layer and calculates point sources using the following series of equations applied to the four nearest speakers:

Below layer:

$$BL_{amp} = cos(BL_{distance} * \pi/2) * cos(B_{amp} * \pi/2) \quad (1)$$

$$BR_{amp} = sin(BL_{distance} * \pi/2) * cos(B_{amp} * \pi/2) \quad (2)$$

Above layer:

$$AL_{amp} = cos(AL_{distance} * \pi/2) * cos(A_{amp} * \pi/2) \quad (3)$$

$$AR_{amp} = sin(AL_{distance} * \pi/2) * cos(A_{amp} * \pi/2) \quad (4)$$

In the aforesaid equations B stands for the nearest layer below the point source's elevation and A the nearest layer above. BL stands for the nearest left speaker on the below layer, BR for nearest right speaker on the below layer, AL for the nearest left speaker on the above layer, and AR for the nearest right on the above layer. Amp refers to the amplitude expressed as a decimal value between 0 and 1. Distance is a normalized distance between two neighboring speakers expressed as a decimal value between 0 and 1.

The distance is expressed in spherical degrees from the center of the source and loudspeaker position with the ear level being 0° elevation and 0° azimuth being arbitrarily assigned in respect to venue's preferred speaker orientation.

LBAP focuses on the use of a minimal number of speakers. For point sources it can use anywhere between one to four speakers. Arguably its greatest strength resides in its ability to accommodate just about any speaker configuration, from monophonic to as many loudspeakers as the hardware can handle, in perimeter-based spatialization with minimal CPU overhead, requiring only calculation for the affected speakers. The algorithm is further described in greater detail in [17].

Similar to VBAP's Source Spread, LBAP also offers Radius option that accurately calculates per-speaker amplitude based on spherical distance from the point source. It also introduces a unique feature called Spatial Mask (further discussed below). When coupled with the D^4 library, LBAP is further enhanced by a series of unique affordances, including Motion Blur and a battery of time-based editors that leverage oculocentric user interfaces for generating, importing, and manipulating multidimensional data.

D^4 library focuses on mostly open source (MOSS) lean implementation that leverages maximum possible amount of built-in Max objects. This design choice comes with unique challenges, like the lack of graceful handling of determinacy within a Max's multithreaded environment (e.g. using a poly object for dynamic instantiation of per-speaker mask calculating abstractions). It also provides opportunities for the user to build upon and expand library's functionality, thus minimizing the limitations typically associated with closed (a.k.a. blackbox) alternatives.

Other features aimed at addressing the aforesaid aspirational features include support for a broad array of speaker configurations, dynamic runtime reconfigurability of the speaker setup, user-editable loudspeaker configuration syntax, focus on perimeter-based spatialization without need for special spectral adjustment or per-channel processing beyond amplitude manipulation, low-latency real-time-friendly operation, built-in audio bus system per each audio source designed to promote signal isolation and streamline editing, independent layers (e.g. sub arrays), and focus on leveraging real-world acoustic conditions where vantage point is treated as an asset rather than a hindrance. LBAP does not aim to compensate for vantage point perceptual variances. This is in part because such an implementation is seen as offering opportunities for broadening of cognitive bandwidth by cross-pollinating different modalities (e.g. location-based awareness and aural perception), and also in part because it minimizes the need for idiosyncrasies that may limit system's scalability and transportability, and/or adversely affect its overall CPU overhead. D^4's lean design promotes optimization and scalability, as well as easy expansion, with the ultimate goal of promoting transportability. The library can serve as a drop-in replacement for the mainstream spatialization alternatives that rely on azimuth and elevation parameters. Furthermore, D^4 tools promote ways of retaining time-based spatial configuration in its original, editable format that can be used for real-time manipulation. The same can be also used to render time-based data for different speaker configurations and later playback that bypasses potentially CPU intensive real-time calculation. To aid in this process the system offers tools for playback of pre-rendered spatial data thereby making its playback resolution limited only by the per-loudspeaker amplitude crossfade ramp values, whose primary purpose is to prevent clicks in audio output.

3. UNIQUE AFFORDANCES

3.1 Spatial Mask

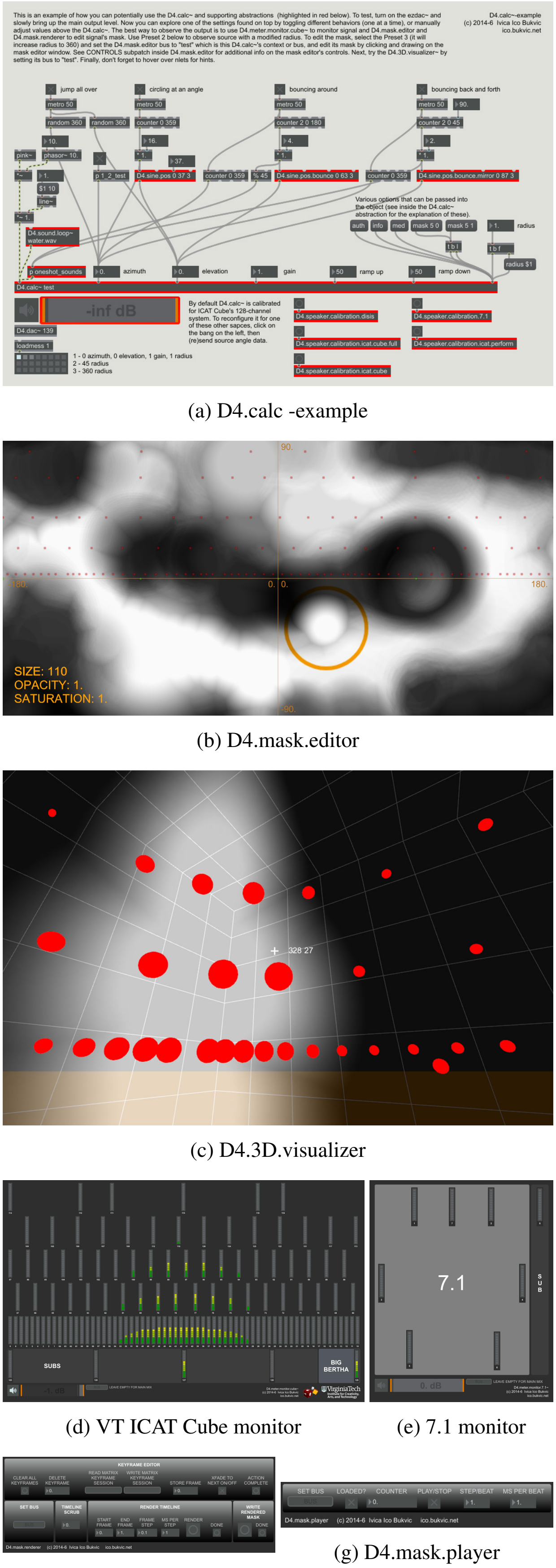
Spatial Mask (SM) is one of the features unique to the D^4 ecosystem. Akin to that of its visual counterpart LBAP considers the entire perimeter space to have the default mask of 1. This means wherever the point source is and whatever its radius, it will populate as many loudspeakers as its computed amplitude and radius permit based solely on its calculated amplitude curve. The spatial mask, however, can be changed with its default resolution down to 0.5° horizontally and 1° vertically, giving each loudspeaker a unique maximum possible amplitude as a float point value between 0 and 1. As a result, a moving source's amplitude will be limited by loudspeaker's corresponding mask value as it traverses the ensuing loudspeaker. This also allows a situation where a point source with 180° radius that emanates throughout all the loudspeakers can now be dynamically modified to map to any SM, thus creating complex shapes that go well beyond the traditional spherical sources.

3.2 Time-Based Editing Tools

SM implementation leverages Jitter library and all its affordances, making it convenient to import and export SM snapshots and automate time-based alterations. Akin to a single channel video, D^4's SM editing tools use grayscale 2D matrix to calculate the ensuing per-loudspeaker mask. As of version 2.1.0, the time-based editing tools allow for SM translation and can couple azimuth, elevation, as well as Motion Blur data into a single coll-formatted file that is accompanied by matrices corresponding with each keyframe. The library can then interpolate between those states at user-specified resolution in real-time and via batch rendering, allowing for time-stretching and syncing with content of varying duration.

The entire D^4 ecosystem is virtual audio bus aware and widgets (where appropriate) can be easily reconfigured to monitor and/or modify properties of a specific bus. Leaving the bus name blank will revert to monitoring main outs (where applicable). Apart from the D4.calc abstraction that encompasses library's core functionality and instantiates a single movable bus source, the supporting tools also include (Figure 1):

- **D4.mask.editor** (a.k.a. the editor) designed to provide a basic toolset for visual mask editing. It leverages Jitter library to provide SM painting ability and link it with a particular bus or a sound source;



This is an example of how you can potentially use the D4.calc~ and supporting abstractions (highlighted in red below). To test, turn on the ezdac~ and slowly bring up the main output level. Now you can explore one of the settings found on top by toggling different behaviors (one at a time), or manually adjust values above the D4.calc~. The best way to observe the output is to use D4.meter.monitor.cube~ to monitor signal and D4.mask.editor and D4.mask.renderer to edit signal's mask. Use Preset 2 below to observe source with a modified radius. To edit the mask, select the Preset 3 (it will increase radius to 360) and set the D4.mask.editor bus to "test" which is this D4.calc~'s context or bus, and edit its mask by clicking and drawing on the mask editor window. See CONTROLS subpatch inside D4.mask.editor for additional info on the mask editor's controls. Next, try the D4.3D.visualizer~ by setting its bus to "test". Finally, don't forget to hover over nlets for hints.

D4.calc~-example
(c) 2014-6 Ivica Ico Bukvic
ico.bukvic.net

Various options that can be passed into the object (see inside the D4.calc~ abstraction for the explanation of these).

By default D4.calc~ is calibrated for ICAT Cube's 128-channel system. To reconfigure it for one of these other sapces, click on the bang on the left, then (re)send source angle data.

1 - 0 azimuth, 0 elevation, 1 gain, 1 radius
2 - 45 radius
3 - 360 radius

(a) D4.calc -example

(b) D4.mask.editor

(c) D4.3D.visualizer

(d) VT ICAT Cube monitor (e) 7.1 monitor

(g) D4.mask.player

(f) D4.mask.renderer

Figure 1: A collection of D^4's widgets.

- **D4.3D.visualizer** that allows users to monitor both SM and the final bus output in a spatially aware 3D environment;
- **D4.meter.monitor.*** is a collection of abstractions that offer a more traditional way of monitoring levels. They are built using D^4's helper abstractions designed to promote rapid-prototyping of configurations other than the ones already included with the library. As of version 2.1.0, the library offers visualizer for three Virginia Tech signature spaces and a prototype for 7.1 surround sound system;
- **D4.speaker.calibration.*** provide calibration settings for a growing number of venues, as well as more common multichannel configurations (e.g. 7.1);
- **D4.mask.renderer** (a.k.a. renderer) is the nexus for all time-based editing and rendering. All of the aforesaid tools, including the D4.calc abstraction are designed to interface with this object, feed edited data and update their own state based on the data provided by the renderer, and
- **D4.mask.player** that can play data rendered by the D4.mask.renderer and feed it into the target D4.calc (a.k.a. bus).

The entire library is envisioned as a modular collection of self-standing, yet mutually aware widgets. As a result, user can customize their workspace as they deem fit. The same widgets can be also embedded as GUI-less abstractions in their own patches by leveraging the annotated inlets and outlets, as well as included documentation and examples. In addition, due to their MOSS design the widgets themselves can be further enhanced (e.g. by altering the default speaker configuration that is preloaded within each D4.calc , adding custom filters to specific outputs, or by introducing new and more advanced ways of processing SM matrices). The enhancements that prove particularly useful may be eventually merged into the future upstream releases.

The ability of each widget to be utilized independently from others is limited only by context. For instance, editing SM makes no sense unless the bus being edited actually exists. Likewise, storing SM is impossible without having a renderer monitoring the same bus. To maximize the possible number of viable configurations has required some widgets to carry redundant implementations. For instance, editor if used solely to experiment on a particular bus without the intent to store the ensuing SM (e.g. for a real-time manipulation), requires D4.mask.calculator abstraction that is also present within the renderer. Consequently, to minimize the redundancy and the ensuing CPU overhead in situations where both abstractions are present within the same bus pipeline, the library has a framework to autodetect such a condition and minimize the redundancy by disabling the calculator within the editor and forwarding the editor data directly to the renderer.

3.3 Helper Abstractions

In addition to the aforesaid widgets, D^4 also offers a collection of helper abstractions designed to streamline library's utilization in more complex scenarios. D4 , D4.dac , and D4.cell are abstractions used for the dynamic creation of the bus outputs, as well as main outputs, both of whose state can be monitored and manipulated (e.g. bus up- and down-ramps that can be used to manipulate moving source attack and trail envelope, effectively resulting in the aural equivalent of the motion blur (MB) [17]). D4.meter.cell is designed to be used primarily as a Max's visual abstraction (a.k.a. bpatcher) for the purpose of rapid prototyping spatially-aware visual level monitors, whereas D4.meter.3D.cell is used for monitoring levels and forwarding those to the D4.mask.3D.visualizer. The library also includes a series of javascripts for managing dynamic generation of necessary buses and outputs. Other smaller convenience abstractions include oneshot audio events (D4.sound.oneshot) and audio loops (D4.sound.loop).

D4.sine.pos* collection of abstractions provide a more advanced automated spatialized source motion. As shown in the introductory D4.calc -example patch, when connected to the D4.calc 's azimuth and elevation values, these abstractions can provide circular motion at an angle other than the traditional horizontal trajectory. *bounce version mimics a bouncing object, while *mirror variant enables bouncing against both extremes. Each of the abstractions can take optional arguments that modulate the range and offset.

4. CONCLUSIONS AND FUTURE WORK

D^4 is a rapidly growing Max library designed to address the limited transportability of spatial audio using HDLAs in artistic and reseach contexts. It does so by coupling a new Layer Based Amplitude Panning algorithm with a battery of supporting time-based tools for importing, editing, exporting, and rendering spatial data, including real-time low-latency HDLA scenarios. The newfound affordances, such as the Radius, Spatial Mask, and Motion Blur, when combined with Jitter-based editing tools, offers opportunities for researching new approaches to audio spatialization. These include scientific research that furthers understanding of human spatial perception and more importantly leveraging the ensuing knowledge for the purpose of emancipating spatial audio dimension both within the artistic and research contexts.

Given D^4's increasing complexity, it is possible that its optimal implementation may in future require a self-standing application. As of right now, it is unclear whether the current MOSS approach as a Max library will prove an environment conducive of creativity it aims to promote, particularly in respect to a battery of tools and widgets that in their current form defy traditional appearances of time-based editing tools. It is author's intention to continue investigating optimal ways of introducing timeline-centric features within the existing implementation and consider expanding to other platforms based on user demand.

5. OBTAINING D^4

D^4 can be downloaded from `http://ico.bukvic.net/main/d4/`.

6. REFERENCES

[1] G. Kramer, *Auditory display: Sonification, audification, and auditory interfaces*. Perseus Publishing, 1993. [Online]. Available: http://dl.acm.org/citation.cfm?id=529229

[2] V. Pulkki, "Virtual sound source positioning using vector base amplitude panning," *Journal of the Audio Engineering Society*, vol. 45, no. 6, pp. 456–466, 1997. [Online]. Available: http://www.aes.org/e-lib/browse.cfm?elib=7853

[3] N. Barrett, "Ambisonics and acousmatic space: a composers framework for investigating spatial ontology," in *Proceedings of the Sixth Electroacoustic Music Studies Network Conference*, 2010. [Online]. Available: http://www.natashabarrett.org/EMS_Barrett2010.pdf

[4] B. Carty and V. Lazzarini, "Binaural HRTF based spatialisation: New approaches and implementation," in *DAFx 09 proceedings of the 12th International Conference on Digital Audio Effects, Politecnico di Milano, Como Campus, Sept. 1-4, Como, Italy*. Dept. of Electronic Engineering, Queen Mary Univ. of London,, 2009, pp. 1–6. [Online]. Available: http://eprints.maynoothuniversity.ie/2334

[5] V. Pulkki, "Virtual sound source positioning using vector base amplitude panning," *Journal of the Audio Engineering Society*, vol. 45, no. 6, pp. 456–466, 1997. [Online]. Available: http://www.aes.org/e-lib/browse.cfm?elib=7853

[6] T. Lossius, P. Baltazar, and T. de la Hogue, "DBAPdistance-based amplitude panning." Ann Arbor, MI: Michigan Publishing, University of Michigan Library, 2009. [Online]. Available: http://www.trondlossius.no/system/fileattachments/30/original/icmc2009-dbap-rev1.pdf

[7] Z. Seldess, "MIAP: Manifold-Interface Amplitude Panning in Max/MSP and Pure Data," in *Audio Engineering Society Convention 137*. Audio Engineering Society, 2014. [Online]. Available: http://www.aes.org/e-lib/browse.cfm?conv=137&papernum=9112

[8] K. Brandenburg, S. Brix, and T. Sporer, "Wave field synthesis," in *3DTV Conference: The True Vision-Capture, Transmission and Display of 3D Video, 2009*. IEEE, 2009, pp. 1–4. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5069680

[9] M. Puckette, "Max at seventeen," *Computer Music Journal*, vol. 26, no. 4, pp. 31–43, 2002. [Online]. Available: http://www.mitpressjournals.org/doi/pdf/10.1162/014892602320991356

[10] ——, "Pure Data: another integrated computer music environment," *IN PROCEEDINGS, INTERNATIONAL COMPUTER MUSIC CONFERENCE*, pp. 37–41, 1996. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.41.3903

[11] M. Kronlachner, "Ambisonics plug-in suite for production and performance usage," in *Linux Audio Conference*. Citeseer, 2013, pp. 49–54. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.654.9238&rep=rep1&type=pdf#page=61

[12] J. C. Schacher, "Seven years of ICST Ambisonics tools for maxmsp?a brief report," in *Proc. of the 2nd International Symposium on Ambisonics and Spherical Acoustics*, 2010. [Online]. Available: http://ambisonics10.ircam.fr/drupal/files/proceedings/poster/P1_7.pdf

[13] "Sound Particles - Home," http://www.sound-particles.com/.

[14] "D-Mitri : Digital Audio Platform | Meyer Sound," http://www.meyersound.com/product/d-mitri/spacemap.htm. [Online]. Available: http://www.meyersound.com/product/d-mitri/spacemap.htm

[15] C. Ramakrishnan, "Zirkonium: Non-invasive software for sound spatialisation," *Organised Sound*, vol. 14, no. 03, pp. 268–276, Dec. 2009. [Online]. Available: http://journals.cambridge.org/article_S1355771809990082

[16] E. Corteel, A. Damien, and C. Ihssen, "Spatial sound reinforcement using Wave Field Synthesis. A case study at the Institut du Monde Arabe," in *27th TonmeisterTagung-VDT International Convention*, 2012. [Online]. Available: http://www.wfs-sound.com/wp-content/uploads/2015/03/TMT2012_CorteelEtAl_IMA_121124.pdf

[17] I. I. Bukvic, "3D TIME-BASED AURAL DATA REPRESENTATION USING D 4 LIBRARY'S LAYER BASED AMPLITUDE PANNING ALGORITHM," in *the 22nd International Conference on Auditory Display (ICAD 2016). July 3-7, 2016, Canberra, Australia*, 2016. [Online]. Available: http://www.icad.org/icad2016/proceedings2/papers/ICAD2016_paper_10.pdf