



Audio Engineering Society Convention Paper 9112

Presented at the 137th Convention
2014 October 9–12 Los Angeles, USA

This Convention paper was selected based on a submitted abstract and 750-word precis that have been peer reviewed by at least two qualified anonymous reviewers. The complete manuscript was not peer reviewed. This convention paper has been reproduced from the author's advance manuscript without editing, corrections, or consideration by the Review Board. The AES takes no responsibility for the contents. Additional papers may be obtained by sending request and remittance to Audio Engineering Society, 60 East 42nd Street, New York, New York 10165-2520, USA; also see www.aes.org. All rights reserved. Reproduction of this paper, or any portion thereof, is not permitted without direct permission from the Journal of the Audio Engineering Society.

MIAP: Manifold-Interface Amplitude Panning in Max/MSP and Pure Data

Zachary Seldess, Senior Audio Research Engineer

Sonic Arts R&D, Qualcomm Institute, Calit2, UC San Diego, San Diego, USA
zseldess@gmail.com

ABSTRACT

This paper discusses MIAP (Manifold-Interface Amplitude Panning), a new freely available implementation of Meyer Sound's SpaceMap abstract spatialization software via a collection of C externals for Max/MSP and Pure Data. SpaceMap's technical and conceptual innovations are discussed and placed within the larger context of widely available codified spatialization algorithms and approaches such as Vector-base amplitude panning. An examination of the new implementation is made along with discussion of added features resulting from the translation.

1. INTRODUCTION

Common sound spatialization tools and techniques, such as Vector-base amplitude panning and Ambisonics, have traditionally provided methods for composers and sound designers to author spatial sound as a virtual extension of a listener's real space. There has been much past academic writing on such tools and approaches over the last twenty years, both in dissertation and scholarly article formats [1-9, among many others]. However, spatialization approaches that deal in more flexible abstracted source-to-speaker and speaker-to-world relationships, approaches that make no carved-in-stone assumptions about listener position and physical placement and orientation of loudspeakers, nor about the precise manner in which a sound will manifest in real-space via speakers (i.e. as a point-source, diffuse field, etc.) are neither extensively discussed in the

academic literature, nor are there many readily available tools in popular use by composers and sound designers.

In 1986, Steve Ellison, formerly of LCS Audio, now Meyer Sound Laboratories Inc.¹, invented a novel amplitude panning algorithm using barycentric coordinates to derive equal power speaker gains among collections of triplets of loudspeakers. Around the time of invention, Ellison was working on a real-time computer music composition system for a poly-media performance space built by "Floating Exceptions", a collective of computer artists in Canberra, Australia

¹ Meyer Sound, SpaceMap, CueStation, Matrix3, and D-Mitri are trademarks of Meyer Sound Laboratories, Incorporated. Meyer Sound, SpaceMap and D-Mitri are registered in the U.S. Patent & Trademark Office.

[10]. For this project, Ellison needed a computationally efficient method to compute gains for loudspeakers placed on the surface of a custom-built 16-channel geodesic dome. The triangular placement of speakers at nodes on the dome seemed to dictate his decision to pan with triplets rather than other groupings of speakers, and out of that practical constraint, he realized the significance and relative computational simplicity of barycentric coordinates as a method for computing power-preserving loudspeaker gains².

Since the algorithm's invention and first implementation, the software, originally called SpaceNodes, later renamed SpaceMap [12], has undergone many refinements and iterations, with later versions used for live sound reinforcement to control analog matrix mixers, and subsequently implemented on three generations of digital audio mixers. Over the two decades that followed the software's first commercial iteration in the early 1990s, the authoring and control constructs and graphical user interface underwent several practically-driven improvements (such as the taxonomy of "nodes" into one of four possible types, to be discussed later in paper) that transformed a simple barycentric panning algorithm into a powerful and elegant solution for large-scale abstract spatial sound design [10] [13]. SpaceMap is unique as a spatialization tool in that:

- It is essentially unknown outside of the large-scale live theatrical sound design community, particularly within scholarly communities working in spatial sound tools and techniques. To date, the only scholarly mention of Ellison's status as inventor of barycentric panning, as well as the resulting software built around that algorithm, is found in a single sentence of Ville Pulkki's 2001 dissertation on Vector-base amplitude panning, citing verbal correspondence with Ellison as the source [2].
- It is perhaps the only extant example of a stable, codified tool for spatial sound design whose algorithm and functionality have developed entirely out of practical necessities, as opposed to more

² Ellison's use of the barycentric coordinate system was a purely intuitive, albeit elegant, solution. At the time of invention, he didn't have a name for the technique. He learned sometime later that August Ferdinand Möbius had in fact introduced the underlying math in 1827 [11].

academic notions of "what would be ideal".

Though SpaceMap can easily be used to control many "idealized" rendering configurations such as hemispherical speaker domes and standard surround sound layouts, it is first and foremost a tool to author and control sound within idiosyncratic, site-specific, and often incredibly complex multi-channel systems [10]. Due to its evolution being intimately tied up in supporting and solving for the challenges of complex site-specific speaker layouts, the solutions that Ellison and his colleagues have arrived at over the years, elegant and simple as they appear to be in retrospect, provide a powerful example of the potential benefits of use-case driven software design.

In this paper, I will explain and illustrate the primary features of SpaceMap within Meyer Sound's CueStation control software for the Matrix3 and D-Mitri digital show control systems [13]. I will then dedicate the remainder of the paper to describing the MIAP externals: a new, freely available implementation of SpaceMap by the author for Max/MSP [14] and Pure Data [15]. I will discuss important functionality ported to the externals, describing new features added as a result of the port. I will also briefly discuss potential future iterations and improvements to the objects. Finally, in an effort to provide a tangible way to learn about SpaceMap and the creative sound design possibilities that it inspires, I will provide information for the wider community to freely access and use these externals within Max/MSP and Pd.

2. THE SPACEMAP APPROACH

2.1. Non-Centricity

Sound spatialization approaches, such as Vector-base amplitude panning and Ambisonics, treat the authoring of spatial sound essentially as an act of augmenting the aural environment of an idealized listening area, or "sweet-spot". This *centric* model of conceptualizing and controlling spatial sound, where a sound source is placed in virtual space around an abstract idealized origin, has a great many advantages, as exemplified by the popularity of the approach in the domains of audio for computer games [16] and virtual reality [17], as well as the potential adoption of object-based audio for cinema sound [18][19]. Perhaps the most significant advantage of the centric model is its implied generalization of the sound space.

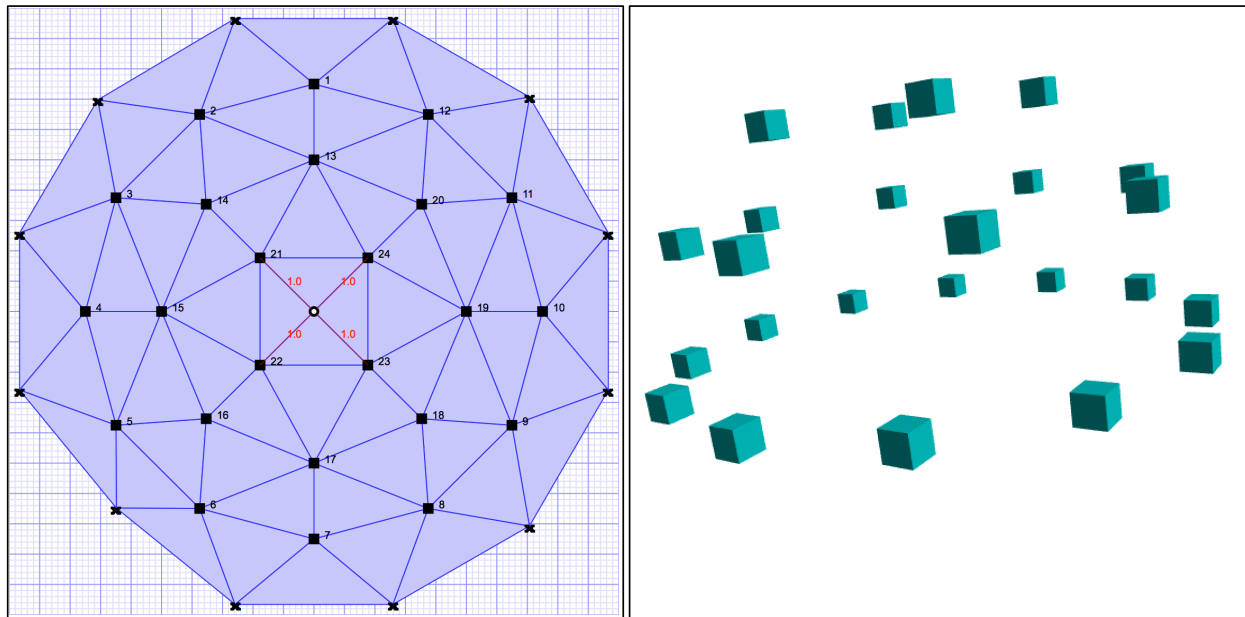


Figure 1: A birds-eye manifold representation (left) of a 24-channel hemispherical loudspeaker configuration (right). Square nodes in the map on the left represent individual loudspeakers on the right.

That is, spatial centrality provides a vehicle for authoring sound without requiring that the author consider the mechanisms by which the sound will be realized (i.e. number of loudspeakers, size of the room, etc.). By essentially removing the physical rendering mechanism from the language of spatial control (i.e. referring to the sound in 2D or 3D Cartesian or Polar coordinates), one maximizes the scalability and portability of the sound, providing a vehicle for that sound to be easily translated across a variety of rendering configurations (e.g. headphones, 5.1, 7.1, 22.2, etc.).

Although the centric approach to conceptualizing spatial sound makes good sense across a wide variety of use-cases, it does rely heavily on a set of assumptions to be met in order for the rendered results to conform to authored expectations. Two such important assumptions are:

1. All loudspeakers must be equidistant from the assumed listening area (though this can be practically remedied by applying appropriate per-speaker delays, attenuation, and/or EQ),
2. All loudspeakers must be angled in towards a converging listener “sweet-spot”.

The latter point is perhaps the most critical assumption upon which spatialization approaches that employ centric models rely. Both assumptions, though often adhered to in controlled environments such as computer music studios and virtual reality labs, also happen to be quite commonly *in conflict* with the realities of loudspeaker setups for a great many real-world applications, such as can be found in the realms of site-specific sound art installation, and large-scale theatrical sound spectacle³. Environments such as these, where no assumptions can be made about the existence of a single idealized listening area, greatly benefit from the use of a *non-centric* spatial model that, rather than ignore the features of a given installation, embed and foreground those features within the very language of the tool. At least 20 years worth of sound designers have been using SpaceMap for just such non-centric sound spatialization [10].

³ Cirque Du Soleil’s “KA”, for example, employs hundreds of loudspeakers positioned throughout the theater walls, ceilings and balconies, as well as speakers in the backs of every seat!

2.2. Spatial Abstraction

Ellison’s solution to providing a useable control interface for non-centric sound spatialization is based on the act of abstracting the spatial relationships between loudspeakers away from real-space, and redefining those relationships on a two-dimensional topological space, also known as a manifold. Two-dimensional manifolds are commonly referred to as *surfaces* or *maps*, hence the appropriate naming of the software as “SpaceMap”.

In his initial efforts providing a control surface for panning between speakers positioned at the nodes of a geodesic dome [10], Ellison treated this mapped space as a two-dimensional projection of the real three-dimensional space. Such projections of real space onto one or more two-dimensional maps can prove very useful in certain scenarios where the author requires the derivative topological space to visually resemble its real source (see Figure 1). However this approach to abstracting space doesn’t necessarily present an advantage over centric spatialization approaches. For example, the hemispherical speaker configuration depicted in Figure 1 can be supported using VBAP. The significant differentiating factor between the SpaceMap approach and other approaches, including other non-centric algorithms such as DBAP [6] and ViMiC [5], is that of *spatial abstraction*.

From Meyer Sound’s CueStation User Guide:

“A SpaceMap design is constructed from two basic elements: nodes and trisets. *Nodes* commonly represent the positions of loudspeakers or groups of speakers, and may be of several types. *Trisets* link three nodes together, providing the means to distribute signal proportionally among them.” [13] (Figure 2)

In SpaceMap, authors place nodes representing speakers and other entities on two-dimensional maps in arbitrary arrangements. These arrangements may or may not represent a projection of the real-space within which the loudspeakers exist. For example, a map may display two nodes very close together that in real space are the two loudspeakers furthest apart, and vice versa. Additionally, multiple nodes placed throughout a map may refer to the same physical loudspeaker. In short, a map, as defined in SpaceMap, is a flexible control surface upon which speaker-to-speaker relationships are defined without a direct dependency on those speakers’ real-space locations. But in order to understand how this

surface can actually be used, we first need to discuss the basic elements contained within a map.

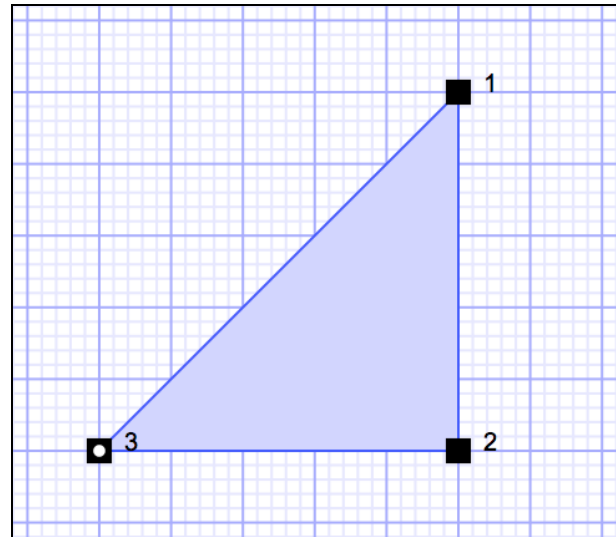


Figure 2: The simplest of maps. Three Speaker nodes linked to form a single Triset.

3. THE ELEMENTS OF SPACEMAP

3.1. Node Types

Nodes on a map represent single speakers (Speaker node), groups of speakers (Virtual node), or points of silence (Silent node). Nodes can also derive their audio from other nodes (Derived nodes). (Figure 3)

3.1.1. Speaker Nodes

Speaker nodes represent single physical outputs in Meyer Sound’s Matrix3 and D-Mitri systems. Those outputs can be connected directly to speakers, or passed to effects processors and other external devices. As mentioned in Section 2.2, multiple speaker nodes in a map can be bound to the same physical output, if desired. Again there is no inherent relationship between a speaker’s physical location in real space and it’s location or locations in mapped space.

3.1.2. Silent Nodes

Silent nodes represent points in the map that, if connected to a Triset, behave similarly to a Speaker node. In contrast to a Speaker node, which sends its

panned signal on to a physical output, signal panned to a Silent node is simply ignored (i.e. silent). Silent nodes provide a convenient way to create fade-ins and fade-outs within a map.

3.1.3. Virtual Nodes

Virtual nodes represent points in the mapped space whose existence is “virtualized” via weighted *links* to one or more of a map’s Speaker nodes. Signal from linked Speaker nodes is, by default, weighted equally between all Speaker nodes, though weightings can be customized to bias the signal of one speaker node over another. Regardless of the proportional link weights applied or the number of Speaker nodes linked to, the resulting signal routed to the linked Speaker nodes will always be power-preserving.

3.1.4. Derived Nodes

Derived Nodes, as the name implies, derive their signal from Speaker nodes. This is the only node type that is not typically part of a Triset, and therefore its physical placement in the mapped space has no functional purpose (aside from visual clarity).

From Meyer Sound’s CueStation User Guide:

“Like a Virtual node it is linked to one or more Speaker nodes, but the logic of the signal routing is reversed. Whereas a Virtual node divides a signal proportionally among its linked Speaker nodes, the Derived node receives a sum of the signals from its linked Speaker nodes.” [13]

Similarly to Virtual nodes, Derived nodes weight the signal received from all linked Speaker nodes equally by default, but this can also be adjusted per-node. In contrast to the Virtual node though, the Derived node receives a linearly summed signal from all Speaker nodes with which it is linked.

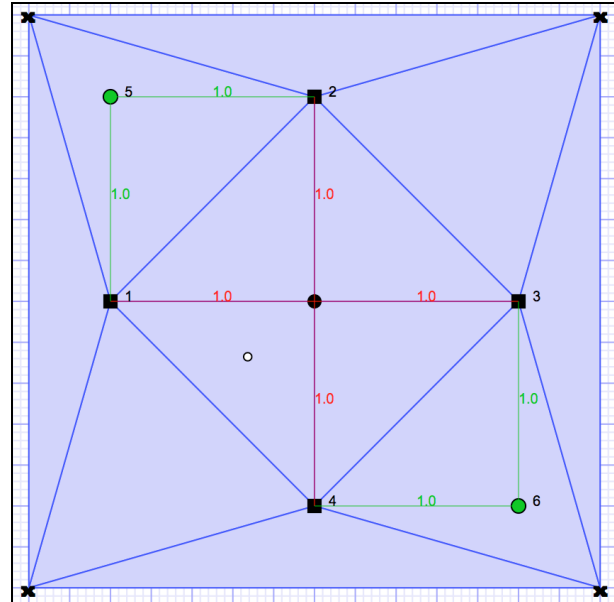


Figure 3: A map employing the four node types. Four Speaker nodes (square) form the primary panning reference points. A Virtual node in the center of the map (black circle) represents a power-preserving distribution to the four Speaker nodes to which it is linked. Two Derived nodes (green circle) receive a linear sum from the Speaker nodes to which they are linked. And Silent nodes (x’s) border the map, providing the ability to gradually fade in and out.

3.2. Trisets and Barycentric Panning

In SpaceMap, *Trisets* are formed out of a combination of three Speaker, Virtual, and/or Silent nodes. Trisets provide a space within which a constant-power panning algorithm can be applied. In SpaceMap, Ellison uses barycentric coordinates to efficiently compute constant-power speaker gains. When dealing with triangles, barycentric coordinates, also known as *areal coordinates*, describe the location of a point P within a triangle ABC via the signed areas of the three sub-triangles PBC , PCA and PAB (Figure 4). Power preserving gains can be derived from barycentric coordinates as follows:

1. Normalize the areas of all sub-triangles so that they sum to 1.
2. Apply the square root of each normalized sub-triangle area as a gain value to the node of the parent triangle not included in that sub-triangle (Figure 4).

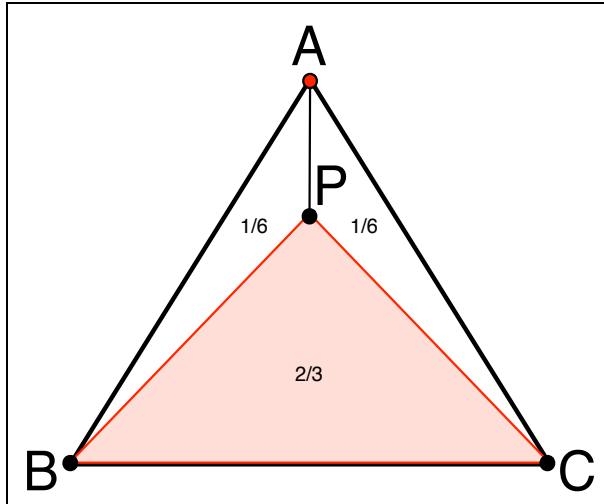


Figure 4: This diagram displays the triangular subdivisions formed by triangle ABC and point P. Normalized areas are displayed for each sub-triangle (1/6, 1/6, 2/3). In SpaceMap, the resulting “gains” applied to each node would be:

$$A = \sqrt{(2/3)}, B = \sqrt{(1/6)}, C = \sqrt{(1/6)}$$

4. MIAP OBJECTS FOR MAX/MSP AND PD

In an effort to expose SpaceMap’s “manifold-interface” approach to a wider audience, and in order to expand on the potential applications of such a non-centric approach, the above functionality has been translated and adapted to the flexible programming environments of Max/MSP and Pure Data via three C externals: a control-rate panner (MIAP), an audio-rate panner (MIAP~), and a UI object (MIAP.map).

4.1. Panning externals: MIAP and MIAP~

The primary difference between MIAP and MIAP~ is that where MIAP outputs a control-rate list of “speaker” gains, which are typically used to adjust a matrix-mixer such as Max’s `matrix~` object⁴, MIAP~ implements the matrix-mixing internally, applying user-defined sample-rate interpolation to the panned input signal. Both panners can be configured to report their “speaker” gains as a list of either linear amplitude or dB values. Using these objects, one can:

- Read, write, and interact with up to two pre-authored maps, stored as JSON files (for Max/Pd-

⁴ For example, Pulkki’s original Max VBAP external outputs control messages for the `matrix~` object.

native use), or in the form of Meyer’s proprietary CueStation project file format,

- Create and edit positions and properties of nodes and trisets,
- Interpolate (i.e. “pan”) between two loaded maps,
- Apply various amounts of power-preserving spread (referred to as “divergence”) among all nodes in a map.

In a few cases, translating the panning functionality to Max/MSP and Pd provided an opportunity to expand on the original implementation. The most notable changes to date are briefly discussed below.

4.1.1. Inter-Map Pan Modes

In SpaceMap, panning between two maps is achieved using a cosine curve for a constant power crossfade. Given the potential in Max and Pd to use a map’s resultant “speaker” gains for interpolation on a wide variety of control points beyond audio signals (e.g. interpolating the parameters of a VST plugin, faders on a lighting board, etc.), it seemed reasonable to provide alternative panning curves. Therefore, for the new externals, users can select from the following “panmodes”: cosine, square root, and linear.

4.1.2. Virtual and Derived Node Types

As described above, in SpaceMap, whereas a Derived node linearly sums the weighted speaker node signals (or gain values for the control-rate panner) with which it is linked, a Virtual node applies a power-preserving mix of the weighted speaker nodes. In the Max/MSP and Pd panner externals, Virtual and Derived nodes may be configured to use either of these summing modalities (power-preserving *or* linear), with the defaults matching that of SpaceMap’s functionality.

4.1.3. Silent Node Weight

In order to provide a more gradual fade when moving toward a Silent node in a Triset, SpaceMap significantly biases the barycentric weight of these node types over other node types, and this Silent node weight is not adjustable by the user. In the Max/MSP and Pd panner externals, Silent node weights default to a value of 1 (i.e. no weight), but a user is allowed to set the relative barycentric weight to any value ≥ 0 .

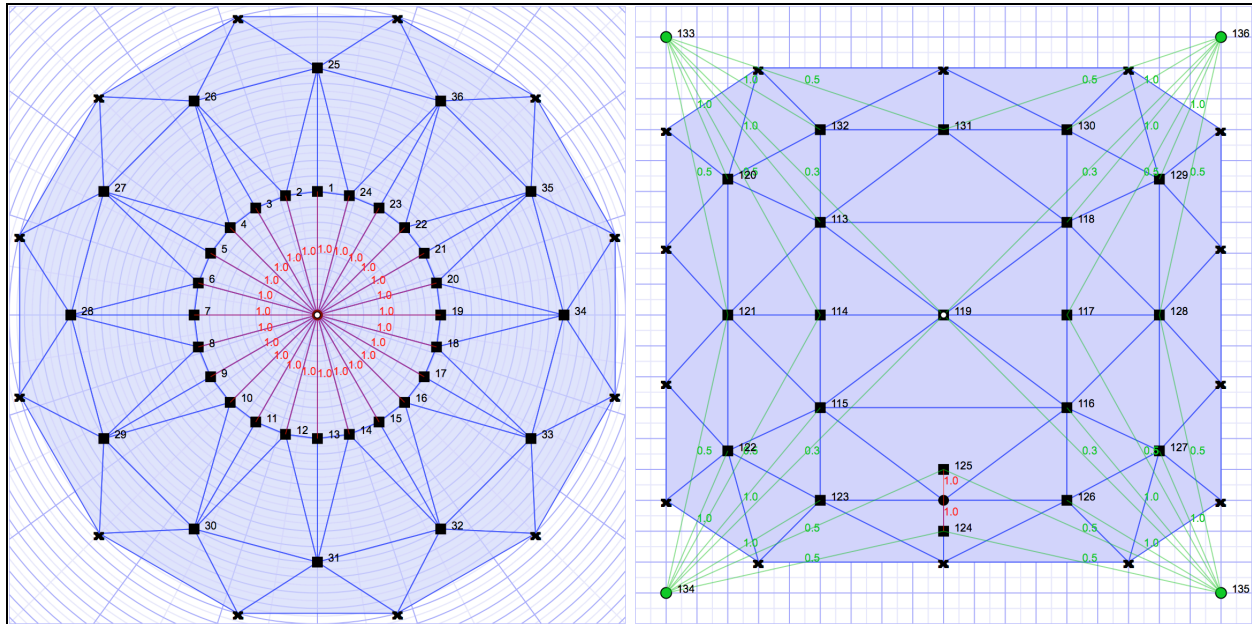


Figure 5: Two example maps rendered using the MIAP.map Max object

4.2. GUI external: MIAP.map

Though the details of a given map are primarily acted upon within the panner objects, for many use-cases it may be preferable that a visual representation of the map be available. The MIAP.map object provides this visual representation of the panning space (Figure 5). Data is shared between panner and UI objects via a common namespace, much like Max/MSP's groove~ and other audio buffer playback objects refer to the namespace of a parent buffer~ object. Using the MIAP.map object, one can:

- Navigate a map loaded into a panner object,
- Visually create and edit positions and certain properties of nodes and trisets,
- Enable/disable the display of various map information such as node and triset IDs,
- Set Virtual and Derived node link weights to display in linear amplitude or dB
- Change the visual appearance of most aspects of the map, such as node and triset border and fill colors.

5. FUTURE WORK

At the time of completing this paper, I have not yet implemented the MIAP.map external for Pure Data. More research into the best approach for the creation of a UI object of this complexity in Pd is required. Also, Ellison's barycentric panning technique need not be limited to sets of three nodes, though trisets make the most sense as a way to group nodes for many map configurations. In some situations however, particularly at the center and edges of a map, it may be beneficial to support sets containing four or more nodes. Therefore, in the future I plan to explore the application of barycentric panning within node sets that form convex polygons of four or more sides. Relatedly, barycentric coordinates may be used to describe subdivisions of three-dimensional convex shapes. Though map creation and control becomes significantly more complex than in the two-dimensional scenario, I hope to explore ways in which this added dimension might add new potential applications for the software.

6. CONCLUSION

In this paper, I have discussed a novel approach to amplitude panning using barycentric coordinates within

triangular sub-regions of a two-dimensional topological surface, or *map*, first invented by Steve Ellison in 1986, and later commercialized and improved within LCS Audio and Meyer Sound's CueStation software for the Matrix3 and D-Mitri digital show control systems. I have also described MIAP: a new implementation of Ellison's "manifold-interface" approach to amplitude panning in the form of three C externals for Max/MSP and Pd. I have detailed some of the changes and additional functionality supported by this new implementation, and listed possible directions for future work. Finally, in hope that this unique approach to spatial sound control may find a wider audience within scholarly and artistic communities, these externals will be made available at the author's website [20].

7. ACKNOWLEDGEMENTS

This work would not have been possible without the generous technical assistance of Steve Ellison, Jeremy Friesner, and others at Meyer Sound Laboratories, Inc.

8. REFERENCES

- [1] Jot et al., "A Comparative Study of 3-d Audio Encoding and Rendering Techniques", Proceedings of the AES 16th International Conference (1999)
- [2] Pulkki, "Spatial Sound Generation and Perception by Amplitude Panning Techniques", Dissertation (2001)
- [3] Neukom, "Ambisonic Panning", Proceedings of the AES 123th Convention, pp. 5–8 (2007)
- [4] Neukom and Schacher. "Ambisonics Equivalent Panning", Proceedings of the 2008 International Computer Music Conference, Belfast, UK (2008)
- [5] Peters et al., "Spatial Sound Rendering in Max/MSP with ViMiC", Proceedings of the 2008 International Computer Music Conference, Belfast, UK (2008)
- [6] Lossius et al., "DBAP - Distance-Based Amplitude Panning", Proceedings of 2009 International Computer Music Conference, Montreal, Canada (2009)
- [7] Kostadinov et al., "Evaluation of Distance Based Amplitude Panning for Spatial Audio" (2010)
- [8] Schacher, "Seven Years of ICST Ambisonics Tools for MaxMSP – A Brief Report", Proceedings of the 2nd International Symposium of Ambisonics and Spherical Acoustics, Paris, France (2010)
- [9] Shin et al., "Efficient 3-D Sound Field Reproduction", Proceedings of the AES 130th Convention, London, UK (2012)
- [10] Ellison, "SpaceMap: 20 Years of Audio Origami", Lighting & Sound America, pp. 80–88 (2013)
- [11] Ellison, Verbal correspondence (2013)
- [12] Ellison, Verbal correspondence (2014)
- [13] Meyer Sound Laboratories Inc., "CueStation 5.2.0 User Guide", pp. 115–141 (2011)
- [14] Zicarelli et al., "Max visual programming environment", <http://cyclimg74.com/products/max> (accessed May 26, 2014)
- [15] Puckette, "Pure Data", Proceedings of the 1996 International Computer Music Conference, San Francisco, USA pp. 224–227 (1996)
- [16] Goodwin, "3-D Sound for 3-D Games - Beyond 5.1", Proceedings of the AES 135th Convention, London, UK (2009)
- [17] Wakefield et al., "The AlloBrain: An Interactive Stereographic, 3D Audio Immersive Environment", CHI Conference Workshop on Sonic Interaction Design (2008)
- [18] Jot, "Next Generation Immersive Audio Content Standard and Consumer Format", IEEE International Symposium on Broadband Multimedia Systems and Broadcasting, Beijing, China (2014)
- [19] Kalker, "Open Object-Based Audio Master Format", ISO/IEC JTC1/SC29/WG11 MPEG2013/M28914, Incheon, Rep. of Korea (2013)
- [20] Seldess, <http://www.zacharyseldess.com>