A versatile workstation for the diffusion, mixing, and post-production of spatial audio

Thibaut CARPENTIER
UMR 9912 STMS IRCAM–CNRS–UPMC
1, place Igor Stravinsky,
75004 Paris, France,
thibaut.carpentier@ircam.fr

Abstract

This paper presents a versatile workstation for the diffusion, mixing, and post-production of spatial sound. Designed as a virtual console, the tool provides a comprehensive environment for combining channel–, scene–, and object–based audio. The incoming streams are mixed in a flexible bus architecture which tightly couples sound spatialization with reverberation effects. The application supports a broad range of rendering techniques (VBAP, HOA, binaural, etc.) and it is remotely controllable via the Open Sound Control protocol.

Keywords

sound spatialization, mixing, post-production, object-based audio, Ambisonic

1 Introduction

This paper presents a port of the *panoramix* workstation to Linux. First, we give a brief presentation of *panoramix* and typical use-cases of this environment. Then, we present some recently added features and discuss the challenges involved with porting the application to the Linux OS.

Panoramix is an audio workstation that was primarily designed for the post-production of 3D audio materials. The needs and motivations for such tool have been discussed in previous publications [Carpentier, 2016; Carpentier and Cornuau, 2016]: *panoramix* typically addresses the post-production of mixed music concerts[1] where the sound recording involves a large set of heterogeneous elements (close microphones, ambient miking, surround or Ambisonic microphone arrays, electronic tracks, etc). During the post-production stage, the sound engineers need tools for spatializing sonic sources (e.g., spot microphones or electronic tracks), encoding and decoding Ambisonic materials, adding artificial reverberation, combining and mixing the heterogeneous sound layers, as well as rendering, monitoring and exporting the final mix in multiple formats. *Panoramix* provides a unified framework covering all the required operations, and it allows to seamlessly integrate all spatialization paradigms: channel-based, scene-based, and object-based audio.
Besides post-production purposes, *panoramix* is also suitable for the diffusion of sound in live events since the audio engine operates in realtime and without latency.[2] Indeed, it has recently been used by sound engineers and computer musicians in order to control the sound spatialization for live productions at Ircam.

2 Architecture

The general architecture of the workstation has been presented in previous work [Carpentier, 2016]. In a nutshell, the *panoramix* signal flow consists of input tracks which are sent to busses dedicated to spatialization and reverberation effects. All busses are ultimately collected into the Master strip, which delivers the signals to the output audio driver. Each channel strip in the workstation comes with a set of specific DSP features.
One major improvement of the new version herein presented is the introduction of "parallel bussing". Namely, this means that each track can be sent to multiple busses in parallel.[3] The benefit of such parallel bussing architecture is

[1]The practical use of the software in such a context has also been demonstrated in the above-mentioned publications, through the case study of an electro-acoustic piece by composer Olga Neuwirth.

[2]Only a few specific DSP treatments may induce a latency, e.g., the encoding of Eigenmike signals (discussed later in this paper). Also there is the irreducible latency of the audio I/O device.

[3]The number of parallel sends is currently restricted to three busses, referred to as A/B/C. In practical mixing situations, it appeared useless to provide more than three sends although there is no technical constraint to increase this limit.

twofold; it allows:
1) to simultaneously produce a mix in multiple formats: tracks can for instance be sent to a VBAP bus and to an Ambisonic bus; both busses are rendered in parallel, with shared settings, and it is fast and easy to switch from one to another e.g., for A/B comparison.
2) to "hybridize" spatialization techniques: for instance, when producing binaural mixes, it is sometimes useful to combine "true" binaural synthesis (or recordings) with conventional stereophony. Adjusting the level of the two parallel busses, the sound engineer can balance between the 3D layer (with well-known binaural artifacts such as timbral coloration, front-back confusions, in-head localization, etc.) and the stereo layer (often considered as more robust and spectrally transparent). Such hybridization appeared especially useful and convincing when producing content intended for non-individual HRTF listening conditions.

Figures 1 and 2 present the signal-flow graph of the tracks and busses respectively. They also exhibit how the signal processing blocks relate to the controllers exposed in the user interface (see also Figure 4 for a general view of this interface).
When parallel bussing is involved, some elements of the depicted audio graph are replicated and run concurrently.

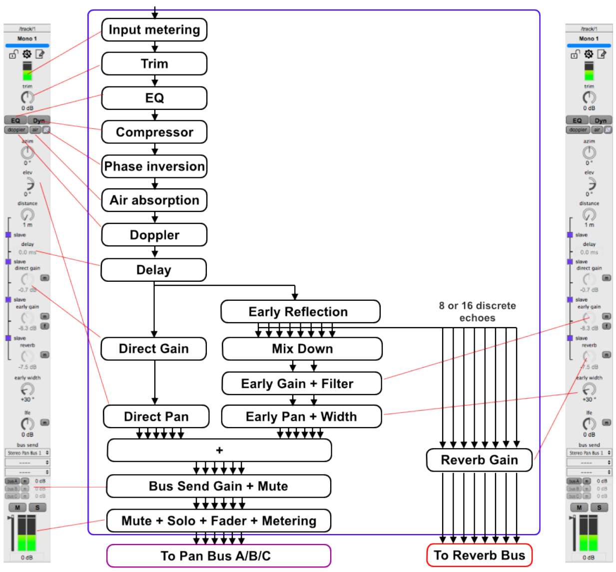

Figure 1: Anatomy of a track: a track is essentially used for pre-processing the incoming audio source (compression, equalization, delay, etc.) and for generating a set of early reflections that will later 1) feed the late reverb FDN and 2) be spatialized.

The overall processing architecture is inspired from the Spat design [Jot and Warusfel, 1995; Jot, 1999; Carpentier et al., 2015] which tightly combines an artificial reverberation engine with a panning module. This framework relies on a simplified space-time-frequency model of room acoustics wherein the generated room effect is divided in four temporal segments (direct sound, early reflections, late reflections, and reverb tail); each segment is individually filtered and then spatialized (direct sound and early reflections are localized as point sources while the late segments are spatially diffuse).
In the first release of *panoramix*, only the filtering of direct sound was proposed. In the presented version, we have introduced additional filters for the early and late reflection sections, therefore extending the range of possible effects.

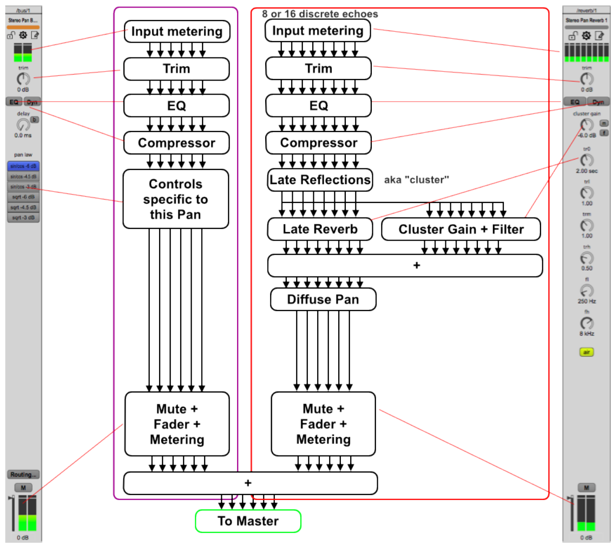

Figure 2: Anatomy of a bus: the purpose of a bus is twofold: it generates a late/diffuse reverberation tail (shared amongst multiple tracks for efficiency) and it provides control over the spatialization rendering. The lefthand side (violet frame) depicts the panning bus; the righthand side (red frame) represents the late reverb bus.

Note that the number of tracks, busses and channel per strip is unlimited, only restricted by the available computing power.

3 Main features

This section presents the main functionalities of the software, with an emphasis on newly added features. The interested reader may also refer to [Carpentier, 2016].

3.1 2D panpot

The first version of *panoramix* was focusing exclusively on 3D rendering approaches, namely VBAP [Pulkki, 1997], Higher Order Ambisonics (HOA) [Daniel, 2001], and binaural [Møller, 1992]. It rapidly appeared convenient to also integrate 2D techniques, as it is common practice to add horizontal-only layers even when mixing for 3D formats. A number of traditional 2D techniques have therefore been implemented (time and/or intensity panning laws such as 2D-VBAP or VBIP [Pernaux et al., 1998], etc). The workstation now offers a broad range of algorithms, being able to address arbitrary loudspeaker layouts.

3.2 Ambisonic processing

Higher Order Ambisonic (HOA) is a recording and reproduction technique that can be used to create spatial audio for circular or spherical loudspeaker arrangements. It has been supported in the workstation since its origin, and further improvements have been made, especially in the encoding and transformation modules.

3.2.1 HOA encoding

Compact spherical microphone arrays such as the Eigenmike[4] are sometimes used for music recordings as they are able to capture natural sound fields with high spatial resolution. The signals captured by such pickup systems do not directly correspond to HOA components; an encoding stage is required. Such encoding usually necessitates to regularize the modal radial filters as they are ill-conditioned for certain frequencies. Various equalization approaches have been proposed in the literature, in particular: Tikhonov regularization [Moreau, 2006; Daniel and Moreau, 2004], soft-limiting [Bernschütz et al., 2011], filter bank applied in the modal domain [Baumgartner et al., 2011]. There is yet no consensus about which method is the most appropriate; consequently they have all been implemented in *panoramix*. An adjustable maximum amplification factor is also controllable by the user.
Besides HOA recordings, it is also possible to synthesize Ambisonic virtual sources and there is no restriction on the maximum encoding order.
Note finally that *panoramix* supports all usual HOA normalization (N3D, N2D, SN3D, SN2D, FuMa, MaxN) and sorting (ACN, SID, Furse-Malham) schemes.

[4] http://www.mhacoustics.com

3.2.2 HOA manipulations

One benefit of the Ambisonic formalism is that a HOA stream can be flexibly manipulated so as to alter the spatial properties of the sound field. In addition to 3D rotations of the sound field [Daniel, 2001; Daniel, 2009], two new transformation operators have been recently integrated to the workstation:
1) a directional loudness processor [Kronlachner and Zotter, 2014] which allows to spatially emphasize certain regions of the sound field (Figure 3) and
2) a spatial blur effect [Carpentier, 2017] which reduces the resolution of an Ambisonic stream, indeed simulating fractional order representation and varying the "bluriness" of the spatial image.
These transformation operators are achieved by applying a (time and frequency independent) transformation matrix in the Ambisonic domain. The implementation is therefore very efficient, making them suitable for realtime automation.

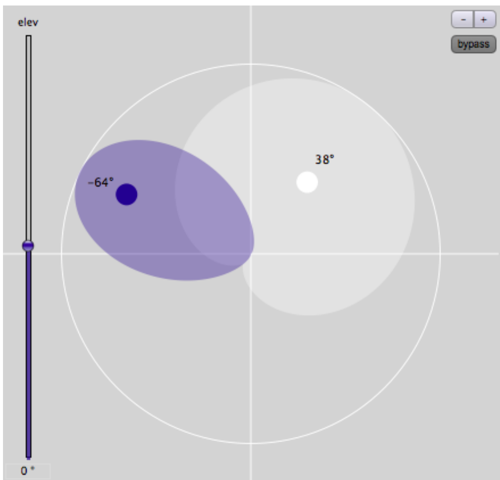

Figure 3: HOA focalization interface: the simple user interface allows to steer one or multiple virtual beams in space; the radial axis is used to control the "selectivity" of the virtual beam (from omnidirectional to highly directional). This is especially useful in post-production contexts, either to emphasize the sound from certain directions (e.g., instruments) or to attenuate undesired regions.

3.2.3 HOA decoding

A HOA bus serves as a decoder (with respect to a given loudspeaker layout) and it comes with a comprehensive set of decoding flavors including: sampling Ambisonic decoder [Daniel, 2001], mode-matching [Daniel, 2001], energy-preserving [Zotter et al., 2012], and all-round decoding [Zotter and Frank, 2012]. In addition, dual-band decoding is possible, with adjustable crossover frequency, and in-phase or max-re optimizations can be applied in each band [Daniel, 2001].

3.3 Binaural rendering

Panoramix implements binaural synthesis for 3D rendering over headphones. It is possible to load HRTF in the SOFA/AES-69 format [Majdak et al., 2013]. Two SOFA conventions are currently supported: "SimpleFreeFieldHRIR" for convolution with HRIR, and "SimpleFreeFieldSOS" for filtering with HRTF represented as second-order sections and interaural time delay.[5]
SOFA data can be either loaded from a local file or remotely accessed through the OpenDAP protocol [Carpentier, 2015a; Carpentier et al., 2014a]. The binaural bus features a user interface for rapid navigation/search through the available SOFA files (Figure 5).

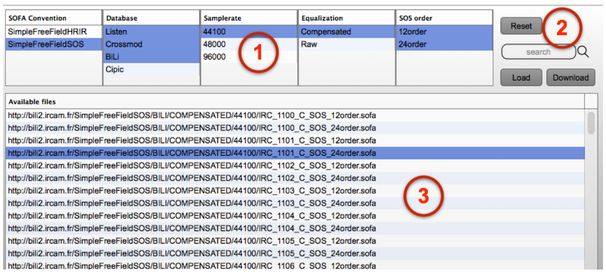

Figure 5: UI for loading or downloading SOFA files. ① Filters for quick search. ② Text search field. ③ Results matching query.

3.4 Reverberation

As mentioned in previous sections, *panoramix* embeds a reverberation engine that allows to generate artificial room effects during the mixing process. The reverb processor currently used is a feedback delay network (FDN) originally designed by [Jot and Chaigne, 1991]. This FDN is particularly flexible and scalable; in typical use-cases, it involves eight feedback channels and provides decay control in three frequency bands.
In addition to that, there is an on-going work to further integrate convolution-based or hybrid reverberators [Carpentier et al., 2014b] in the bus architecture.

[5]see `www.sofaconventions.org` for further details on SOFA conventions.

3.5 OSC communication

All parameters of the *panoramix* application can be remotely accessed via the Open Sound Control (OSC) protocol [Wright, 2005]. This fosters easy and efficient communication with other applications (e.g., Pd) or external devices (e.g., head-tracker for realtime binaural rendering).
OSC communication may also be used for remote automation with a digital audio workstation (DAW) through the ToscA plugin [Carpentier, 2015b]. Note, however, that the latter has not yet been ported to the Linux platform.

A dedicated window allows to monitor the current OSC state of the *panoramix* engine (see ⑦ in Figure 4). Also, the mixing session itself is stored to disk as a "stringified" OSC bundle (human readable and editable).

3.6 Enhanced productivity

A number of other features have been added for enhanced productivity, compared to previous versions. This includes: a large set of keyboard shortcuts (the key mapping can further be customized and stored – see ⑧ in Figure 4) for handling most common tasks (create new tracks, enable/disable groups, etc.), tooltip pop-up that present inline help tips, the possibility to split the console window in multiple windows (especially useful when using multiple screens and dealing with a high number of tracks), etc.

4 Software aspects and Linux port

Panoramix was originally developed as a set of two Max/MSP[6] externals (`panoramix~` for the DSP rendering and `panoramix` for the GUI controller) and released in the form of a Max standalone application for macOS and Windows.
The DSP code is written is `C++`. It is OS-independent, host-independent (i.e. it does not rely on Max/MSP) and highly optimized, extensively using vectorized SIMD instructions and high performance functions from the Intel® Integrated Performance Primitives.[7]
The application can easily handle dozens or even hundreds of tracks on a modern computer.
The GUI component, also written in `C++`, is

[6]`http://www.cycling74.com`
[7]`http://software.intel.com`

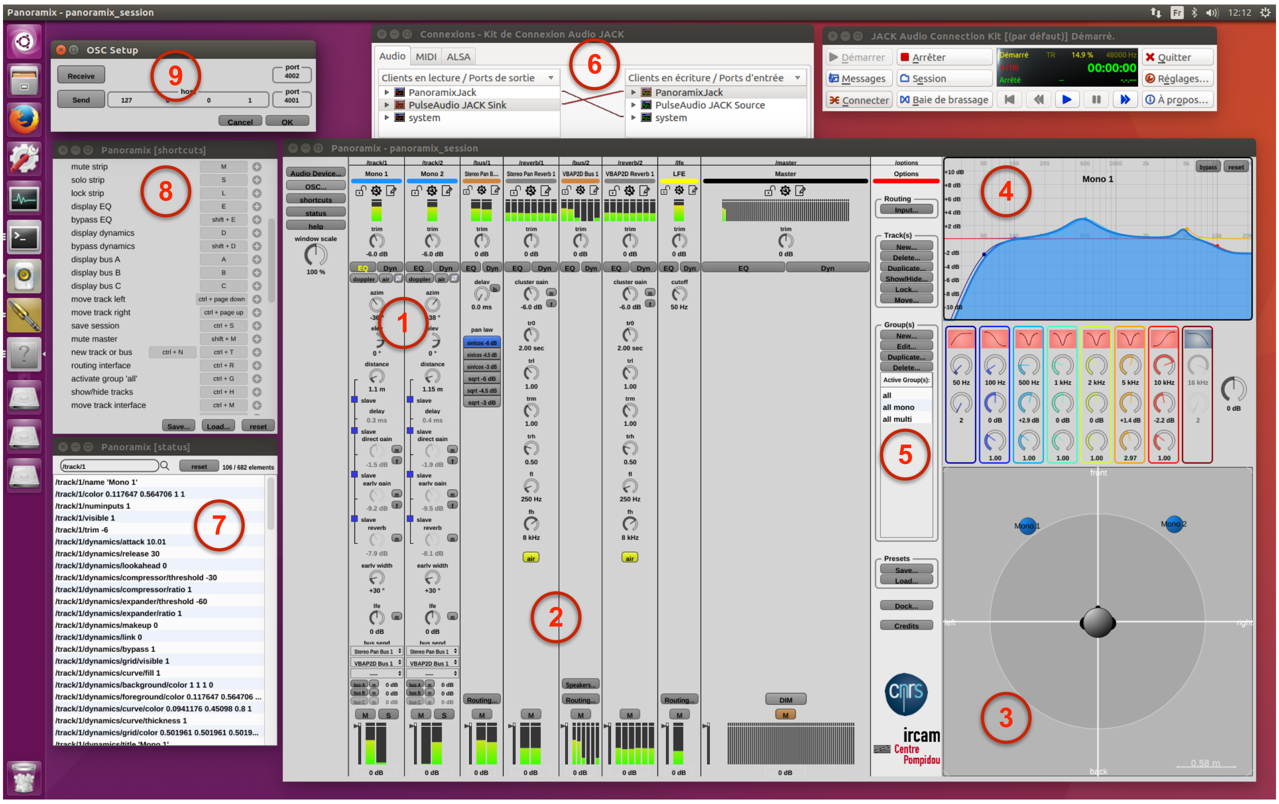

Figure 4: *panoramix* running in Ubuntu with Jack server (QJackctl) as audio pilot. ① Input tracks in the mixing console. ② Spatialization and reverberation busses. ③ Geometrical representation of the sound scene. ④ Parametric equalizer. ⑤ Group management. ⑥ Jack server and inter-application connections. ⑦ Status window: allows to inspect the current state of the engine and all parameters exposed to OSC messaging. ⑧ Shortcut window: allows to edit the key mappings. ⑨ OSC setup window: configure input and output port for remote communication.

built with the Juce[8] framework which facilitates cross-platform development and provides a large set of useful widgets.

For the Linux environment, it was first envisioned to port the Max externals to Pure Data (Pd) [Puckette, 1997]. Porting the DSP engine is straightforward as the Max and Pd APIs are relatively similar in this regard. Porting the GUI object, however, was problematic: Pd uses Tcl/Tk as its windowing system, and to the best of the author's knowledge there is no easy way to embed GUI components developed with other frameworks (such as Juce or Qt) in the Tk engine. As an alternative, it was decided to create an autonomous application, handling both the GUI and the audio engine (i.e. an "AudioAppComponent" in Juce's dialect). The application thus operates independently of any host engine (Pd or Max) and it processes the audio directly to/from the audio devices. Furthermore, it is compatible with the Jack Audio Connection Kit,[9] which makes it pluggable with potentially any audio application. In typical use-cases, a digital audio workstation such as Ardour[10] is used to send audio streams to the *panoramix* processor. The processed buffers may be re-routed to the DAW, e.g., for bouncing, or directly played back through the output device (see Figure 6).

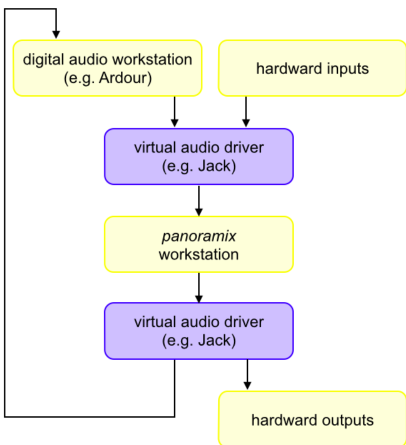

Figure 6: Typical workflow.

[8]http://juce.com/

[9]http://www.jackaudio.org

[10]http://www.ardour.org

5 Conclusions

This paper discussed the Linux port of an audio engine designed for the diffusion, mixing, and post-production of spatial audio. We highlighted several new features that extend the possibilities of the tool and improve productivity and user experience. Future work will mainly focus on the integration of convolution-based reverberation into the framework herein presented.

6 Acknowledgements

The author is very grateful to Clément Cornuau, Olivier Warusfel, and the sound engineering team at Ircam for their invaluable help in the conception of this tool. Thanks also to Anders Vinjar for his insightful advices and beta testing on the Linux platform.

References

Robert Baumgartner, Hannes Pomberger, and Matthias Frank. 2011. Practical implementation of radial filters for ambisonic recordings. In *Proc. of the 1st International Conference on Spatial Audio (ICSA)*, Detmold, Germany, Nov.

Benjamin Bernschütz, Christoph Pörschmann, Sascha Spors, and Stefan Weinzierl. 2011. Soft-limiting der modalen Amplitudenverstärkung bei sphärischen Mikrofonarrays im Plane Wave Decomposition Verfahren. In *Proc. of 37th German Annual Convention on Acoustics (DAGA)*, Düsseldorf, Germany, March.

Thibaut Carpentier and Clément Cornuau. 2016. panoramix: station de mixage et post-production 3D. In *Proc. of the Journées d'Informatique Musicale (JIM)*, pages 162 – 169, Albi, France, April.

Thibaut Carpentier, Hélène Bahu, Markus Noisternig, and Olivier Warusfel. 2014a. Measurement of a head-related transfer function database with high spatial resolution. In *Proc. of the 7th EAA Forum Acusticum*, Kraków, Poland, Sept.

Thibaut Carpentier, Markus Noisternig, and Olivier Warusfel. 2014b. Hybrid Reverberation Processor with Perceptual Control. In *Proc. of the 17th Int. Conference on Digital Audio Effects (DAFx)*, pages 93 – 100, Erlangen, Germany, Sept.

Thibaut Carpentier, Markus Noisternig, and Olivier Warusfel. 2015. Twenty Years of Ircam Spat: Looking Back, Looking Forward. In *Proc. of the 41st International Computer Music Conference (ICMC)*, pages 270 – 277, Denton, TX, USA, Sept.

Thibaut Carpentier. 2015a. Binaural synthesis with the Web Audio API. In *Proc. of the 1st Web Audio Conference (WAC)*, Paris, France, Jan.

Thibaut Carpentier. 2015b. ToscA: An OSC Communication Plugin for Object-Oriented Spatialization Authoring. In *Proc. of the 41st International Computer Music Conference*, pages 368 – 371, Denton, TX, USA, Sept.

Thibaut Carpentier. 2016. Panoramix: 3D mixing and post-production workstation. In *Proc. of the 42nd International Computer Music Conference (ICMC)*, pages 122 – 127, Utrecht, Netherlands, Sept.

Thibaut Carpentier. 2017. Ambisonic spatial blur. In *Proc. of the 142nd Convention of the Audio Engineering Society (AES)*, Berlin, Germany, May.

Jérôme Daniel and Sébastien Moreau. 2004. Further Study of Sound Field Coding with Higher Order Ambisonics. In *Proc. of the 116th Convention of the Audio Engineering Society (AES)*, Berlin, Germany, May.

Jérôme Daniel. 2001. *Représentation de champs acoustiques, application à la transmission et à la reproduction de scènes sonores complexes dans un contexte multimédia.* Ph.D. thesis, Université de Paris VI.

Jérôme Daniel. 2009. Evolving Views on HOA : From Technological To Pragmatic Concerns. In *Proc. of the 1st Ambisonics Symposium*, Graz, Austria, June.

Jean-Marc Jot and Antoine Chaigne. 1991. Digital delay networks for designing artificial reverberators. In *Proc. of the 90th Convention of the Audio Engineering Society (AES)*, Paris, France, Feb.

Jean-Marc Jot and Olivier Warusfel. 1995. A Real-Time Spatial Sound Processor for Music and Virtual Reality Applications. In *Proc. of the of the International Computer Music Conference (ICMC)*, Banff, Canada.

Jean-Marc Jot. 1999. Real-time spatial processing of sounds for music, multimedia and interactive human-computer interfaces. *ACM Multimedia Systems Journal (Special issue on Audio and Multimedia)*, 7(1):55 – 69.

Matthias Kronlachner and Franz Zotter. 2014. Spatial transformations for the enhancement of Ambisonic recordings. In *Proc. of the 2nd International Conference on Spatial Audio (ICSA)*, Erlangen, Germany, Feb.

Piotr Majdak, Yukio Iwaya, Thibaut Carpentier, Rozenn Nicol, Matthieu Parmentier, Agnieszka Roginska, Yôiti Suzuki, Kanji Watanabe, Hagen Wierstorf, Harald Ziegelwanger, and Markus Noisternig. 2013. Spatially Oriented Format for Acoustics: A Data Exchange Format Representing Head-Related Transfer Functions. In *Proc. of the 134th Convention of the Audio Engineering Society (AES)*, Roma, Italy, May 4-7.

Henrik Møller. 1992. Fundamentals of binaural technology. *Applied Acoustics*, 36:171 – 218.

Sébastien Moreau. 2006. *Étude et réalisation d'outils avancés d'encodage spatial pour la technique de spatialisation sonore Higher Order Ambisonics : microphone 3D et contrôle de distance*. Ph.D. thesis, Université du Maine.

Jean-Marie Pernaux, Patrick Boussard, and Jean-Marc Jot. 1998. Virtual Sound Source Positioning and Mixing in 5.1 Implementation on the Real-Time System Genesis. In *Proc. of the Digital Audio Effects Conference (DAFx)*, Barcelona, Spain, Nov.

Miller Puckette. 1997. Pure Data. In *Proc. of the International Computer Music Conference (ICMC)*, pages 224 – 227, Thessaloniki, Greece.

Ville Pulkki. 1997. Virtual Sound Source Positioning Using Vector Base Amplitude Panning. *Journal of the Audio Engineering Society*, 45(6):456 – 466, June.

Matthew Wright. 2005. Open Sound Control: an enabling technology for musical networking. *Organised Sound*, 10(3):193 – 200, Dec.

Franz Zotter and Matthias Frank. 2012. All-round ambisonic panning and decoding. *Journal of the Audio Engineering Society*, 60(10):807 – 820.

Franz Zotter, Hannes Pomberger, and Markus Noisternig. 2012. Energy-preserving ambisonic decoding. *Acta Acustica united with Acustica*, 98:37 – 47.