Twenty Years of Ircam Spat: Looking Back, Looking Forward

Thibaut Carpentier, Markus Noisternig, Olivier Warusfel

Twenty years of Ircam Spat: looking back, looking forward

Thibaut Carpentier, Markus Noisternig, Olivier Warusfel
UMR 9912 STMS IRCAM-CNRS-UPMC
1, place Igor Stravinsky, 75004 Paris
firstname.lastname@ircam.fr

ABSTRACT

Ircam Spat is a software suite dedicated to sound spatialization and artificial reverberation. This paper traces the evolution of the software during the last few years and presents its current state with a focus on the newly available features.

1. HISTORICAL BACKGROUND

Ircam's *Spatialisateur* (commonly referred to as Spat∼) is a tool dedicated to real-time sound spatialization and artificial reverberation. It comes as a library of audio signal processors which core consists of an algorithmic reverberation engine [1] and a panning module. Since its origins in the early 1990s Spat∼ has been conceived as a modular and scalable framework [2, 3] and it is integrated in the Max/MSP [1] environment. Thanks to this modular architecture and design, the tool can be seamlessly adapted for various use cases (live concerts, mixing, post-production, virtual reality, interactive installations, etc.) and configured according to the reproduction system —headphones or loudspeaker arrangement— and the available processing power.

The *Spatialisateur* can be controlled by a high-level interface which includes perceptual features derived from psychoacoustic studies [4, 5]. This allows the user to intuitively specify the characteristics of the desired room effect and to interpolate continuously towards a different acoustic quality, going through natural-sounding transformations.

In terms of software engineering, the *Spatialisateur* was originally developed as a set of Max/MSP patches and abstractions. During the years 1990–2000 the most critical elements (filtering and reverberation stages) were progressively ported to C code as *externals* objects. However this architecture (Max abstractions and a few C *externals*) was not flexible enough for configuring the Spat∼ modules: the *Spatialisateur* was typically restricted to one input source (mono or stereo) and a dozen of output channels. In 2008–2010 it was then decided to completely refactor the software. The refactoring further aimed at taking better advantage of the massively multichannel audio interfaces and computing power now available.

This paper presents an overview of the developments carried out in the last few years, with an emphasis on the new functionalities.

2. SOFTWARE ARCHITECTURE

The *Spatialisateur* has been completely rewritten from scratch in C++ language (and marginally in objective-C). Indeed object-oriented programming is well suited for the modular and scalable design of *Spatialisateur*. The most critical signal processing blocks are highly optimized and vectorized (e.g. with the Accelerate framework [2] under MacOS).

The source code is split in two main libraries: one is dedicated to the signal processing tasks while the other contains the graphical user interfaces. These two libraries are cross-platform, independent from the host application —i.e. they do not rely on Max/MSP— and autonomous (the only dependency is the Juce framework [3], which is used for developing the graphical interfaces).

The benefits from this complete refactoring are manifold:

- the signal processing algorithms show better performances (compared to the patch implementation),
- the algorithms can be flexibly configured: the number of input and output channels of the objects are defined by the @*numinputs* and @*numoutputs* attributes (see Figure 4) and they typically vary from 1 to 512 depending on the setup,
- the *Spatialisateur* no longer relies on Max/MSP and it can spread in new environments: several bindings have been developed in order to cover various use cases (see Figure 1). Max/MSP *externals* are used for concerts or interactive installations. Audio plugins [4] can be inserted in digital audio workstations for mixing and post-production applications. Several tools have been integrated into the OpenMusic environment [6] for spatialization authoring and synthesis. Finally *mexfunctions* are generated for Matlab and they are used (internally) for research and numerical simulation purposes.

In the remainder of this paper, we will focus on the new features of Spat∼ running in Max/MSP as this is the most

[1] http://cycling74.com/

[2] http://developer.apple.com
[3] http://www.juce.com
[4] released by Flux: company http://www.ircamtools.com

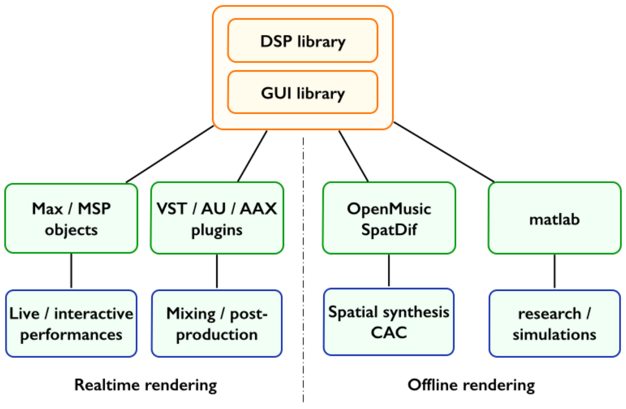


Figure 1. *Spatialisateur* bindings in various working environments.

commonly employed environment and it contains the latest innovations.

3. REVERBERATION ENGINE

Spat∼ is built around a multichannel algorithmic reverberation engine. The generated room effect consists of four temporal segments (Figure 2): the direct sound, a few early reflections generated by a delay line with multiple tap delays, a set of dense late reflections (referred to as *cluster*) generated by multiple delay lines feeding a decorrelation matrix, and a late reverb tail synthesized by a feedback delay network (FDN) [1].

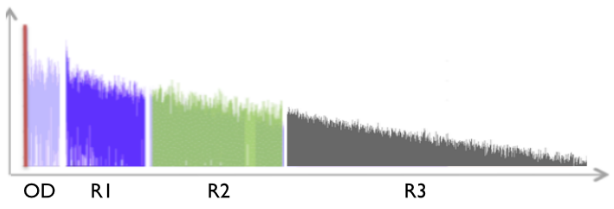


Figure 2. Archetype of an impulse response generated by the *Spatialisateur* reverberation engine. OD: direct sound. R1: early reflections. R2: late reflections. R3: reverberation tail.

With version 4.x of Spat∼ several changes have been applied to the FDN code, compared to the original implementation of Jot [1]: the number of internal feedback channels which was limited to 8 can now be increased to 16 or 32. A higher number of feedback channels allows to increase the modal or echo density; this reveals useful when creating long reverberation effects (several seconds long) or when processing percussive sounds with sharp transients. A set of filters simulating the air absorption have further been inserted into the FDN loop. These filters have a lowpass response which shapes the reverberation profile in a more realistic way. Finally the control over the decay relief (i.e. reverberation time), which was originally operated in two or three frequency bands, has been extended to an arbitrary number of bands (see e.g. the spat.multiverb∼ object).

Besides these evolutions, the global architecture of the reverberation processor (and its link with the panning modules) remains similar to the original design [2, 7]; as a reminder, Figure 3 presents a synoptic view of the audio processing chain.

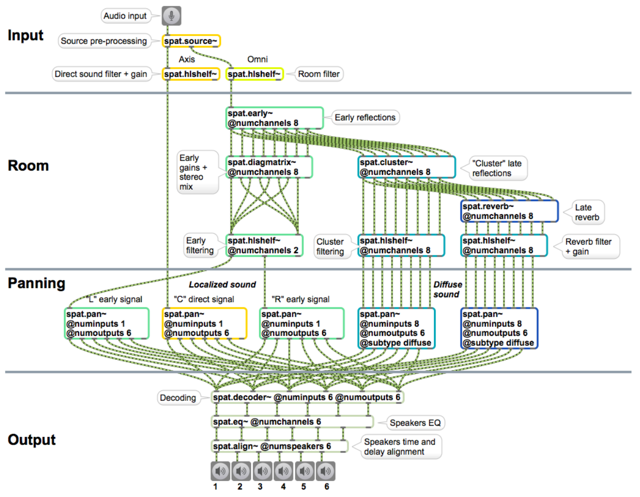

Input
Audio input
Source pre-processing
spat.source~
Axis
Omni
Direct sound filter + gain
spat.hlshelf~
spat.hlshelf~
Room filter
Room
spat.early~ @numchannels 8
Early reflections
Early gains + stereo mix
spat.diagmatrix~ @numchannels 8
spat.cluster~ @numchannels 8
"Cluster" late reflections
spat.reverb~ @numchannels 8
Late reverb
Early filtering
spat.hlshelf~ @numchannels 2
Cluster filtering
spat.hlshelf~ @numchannels 8
spat.hlshelf~ @numchannels 8
Reverb filter + gain
Panning
Localized sound
Diffuse sound
"L" early signal
"C" direct signal
"R" early signal
spat.pan~ @numinputs 1 @numoutputs 6
spat.pan~ @numinputs 1 @numoutputs 6
spat.pan~ @numinputs 1 @numoutputs 6
spat.pan~ @numinputs 8 @numoutputs 6 @subtype diffuse
spat.pan~ @numinputs 8 @numoutputs 6 @subtype diffuse
Decoding
spat.decoder~ @numinputs 6 @numoutputs 6
spat.eq~ @numchannels 6
Speakers EQ
Output
spat.align~ @numspeakers 6
Speakers time and delay alignment
1
2
3
4
5
6

Figure 3. Global architecture of Spat∼. The "Input" module handles the source pre-processing such as filtering and simulating the Doppler effect and air absorption. The "Room" module generates the reverberation effect split in four temporal segments, each filtered independently. These four sections are later panned according to a simple space-time model: early reflections R1 "surround" the direct sound while sections R2 and R3 are spatially diffuse. Finally the "Output" module decodes or transcodes the signals according to the playback system, and it includes loudspeaker equalization and delay/gain alignment.

4. PANNING

From now on, the panning stage is implemented as a unique *external*, spat.pan∼. The object is polymorphic: according to its attributes it configures the appropriate number of input and output channels as well as the type of spatialization to use. A wide range of spatialization algorithms are supported: traditional stereo techniques (AB, XY, MS), binaural synthesis, amplitude panning (e.g. VBAP in 2D or 3D [8], VBIP [9], DBAP [10], SPCAP [11]), etc.

The remainder of this section details some panning families that include the most substantial innovations.

4.1 Higher Order Ambisonics

First order Ambisonic (B-format) encoding and decoding has been supported in the *Spatialisateur* for many years. In the early 2000s, Daniel [12] extended and formalized the Ambisonic paradigm for higher orders (HOA). These scientific advances have been integrated into Spat∼ in the last few years: the spat.pan∼ object now supports 2D and 3D HOA encoding for any order (the algorithms are all implemented with recursive formulae). For instance, pieces created in the performance hall of Ircam (Espace de Projection [13]) typically use 7th order 3D HOA, thus encoding the sound field on 64 spherical harmonics. Near-field Compensation Coding (NFC-HOA, see [14]) is also available in Spat∼ and order-dependent high-pass compensation filters are applied in order to avoid problematic bass-boost when synthesizing finite distance sources. The spat.hoatransform∼ object is used to apply transformations in the spatial Fourier domain; one can e.g. weight the different spherical harmonic components in order to create spatial blur effects, or apply transformation matrix to efficiently rotate the 3D sound scene. Finally spat.decoder∼ performs HOA decoding in 2D or 3D. Traditional decoding approaches consist in sampling the spherical harmonic excitation at the loudspeaker positions or matching the excitation modes of a continuous sound field to those of the loudspeakers. As these techniques perform unsatisfactorily with non-uniformly distributed loudspeaker arrays, a new decoder design has been proposed [15] and implemented in Spat∼. This so-called *energy-preserving* decoder avoids sudden variations in the decoded energy and preserves the apparent width of the virtual sources even with sparse loudspeaker coverage. Furthermore spat.decoder∼ can perform dual-band decoding, with adjustable crossover frequency, and *in-phase* or *max-re* optimizations can be applied in each band [12]. Figure 4 depicts the standard workflow for creating synthetic HOA scenes.

In addition to that, several utility tools have been developed in order to ease frequent operations and to increase the uptake of Ambisonic: spat.hoaconverter∼ converts between various existing formats (N3D, SN3D, FuMa, etc.), spat.eigenencode∼ encodes in the HOA domain the signals captured by a spherical microphone array, spat.hoasorting∼ sorts the HOA channels according to different conventions, etc.

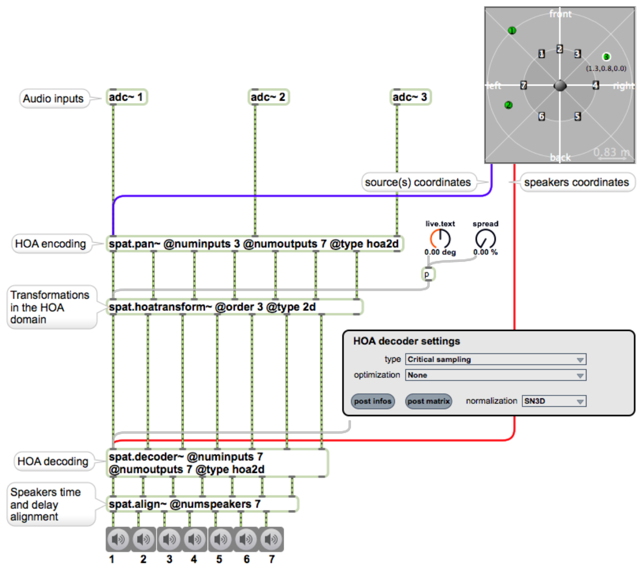


Figure 4. HOA workflow. In this example, three sources are encoded in 2D 3rd order Ambisonic then decoded over seven loudspeakers.

4.2 Nearfield binaural synthesis

Realtime binaural and transaural synthesis have been supported in Spat∼ since its first versions [7]. Such processing techniques rely on filtering the input signals with a set of HRTFs [5] which characterize the transfer path of sound from a position in space to the ear of a listener. HRTFs are usually measured for a large number of positions on a surrounding sphere in the far-field of a listener's head. However, it is well known from literature that HRTFs are significantly different for sources located in the proximal region of the subject. Virtual reality environments constitute a typical use case where binaural synthesis of nearby sources is required. On the one hand, a novel technique has been developed for radial extrapolation of HRTFs [16] and on the other hand, the binaural synthesis renderer (spat.pan∼) has been adapted to sources located in the proximal region. The solution adopted for the realtime implementation is based on non-individualized filters [17] for correcting the ILD [6], cross-ear selection of HRTF filters, and geometrical corrections of monaural gains and delays [18].

4.3 TransPan

TransPan [19] is a new tool dedicated (mainly) to 5.1 mixing and conceived in partnership with the CNSMDP [7]. It aims at overcoming some of the weaknesses of the 5.1 standard in terms of stability and precision of lateral sources. To that purpose, TransPan combines several spatialization layers: 1) a traditional surround panning (constant-power panning and/or multichannel microphone recordings) and 2) a binaural/transaural [7] processing layer using two loudspeaker pairs (L/R and Ls/Rs). The binaural/transaural layer provides the spatial precision that lacks for the lateral images, and can also be employed for creating virtual sources in elevation (outside of the 5.1 horizontal plane).

The spat.transpan∼ object comes as a mixing engine capable of combining the different spatialization layers. The balance (levels, delays and equalization) between the layers can be adjusted manually or automatically based on empirical mixing curves that aim at maximizing the spatial impressions while minimizing the risk of tonal coloration (inherent to any transaural processing). Figure 5 illustrates one view of the spat.transpan∼ control interface.

5. USER INTERFACES

Version 4.x of the *Spatialisateur* library is shipped with several user interfaces for controlling and monitoring sound spatialization.

spat.viewer (see Figure 6) is a tool for the visualization and editing of spatial sound scenes. The object allows to display and manipulate a variable number of sources and loudspeakers in a 2D representation (view from the top, from the bottom, or both side-by-side). The look and feel is highly customizable through Max/MSP attributes. The position and orientation of the scene entities (sources or loudspeakers) can be controlled by a set of Max/MSP messages with a clear, human-readable,

[5] Head-Related Transfer Function
[6] Interaural Level Differences
[7] Conservatoire National Supérieur de Musique et de Danse de Paris

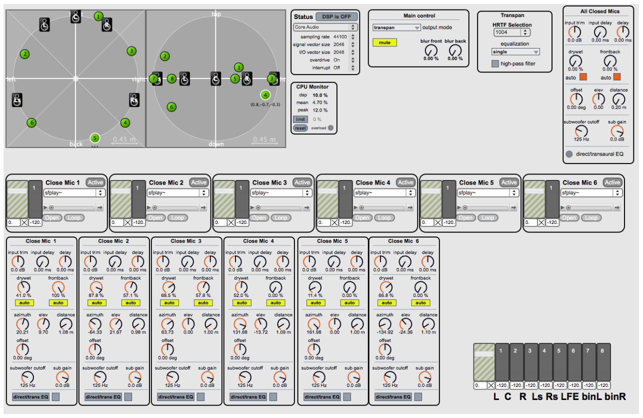


Figure 5. TransPan control interface. Example with six sound sources.

syntax. Many coordinate systems are supported. Furthermore, several utility objects are provided for efficient manipulation of the geometrical data (translation, rotation, scaling, nearest neighbors search, source grouping, conversion between absolute and relative coordinate systems, etc.). It is also worth noting that spat.viewer is not tied to the *Spatialisateur* rendering engine, and it could be used to operate other spatialization processors.

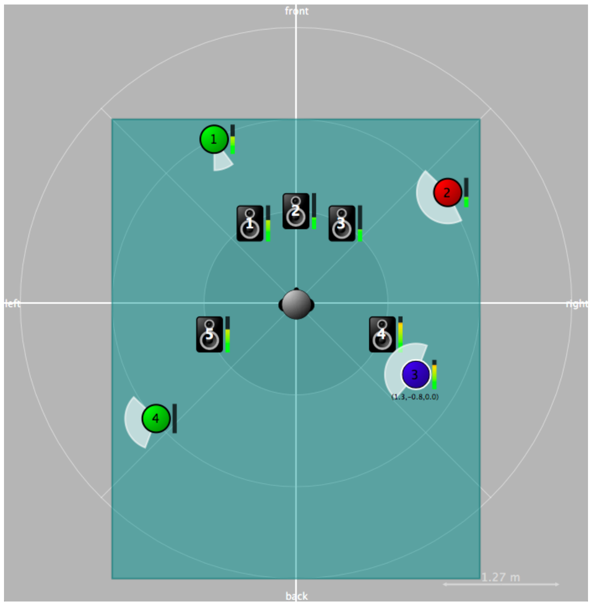
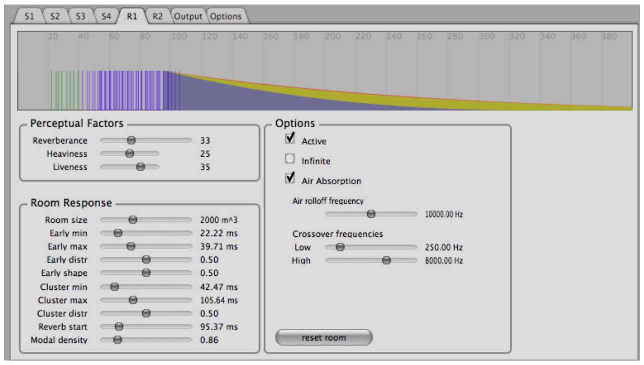


Figure 6. *spat.viewer* interface. Example with four sources, five loudspeakers and one listener, all in a virtual space. View from the top.

spat.oper (see Figures 7 and 8) is the main control interface for the *Spatialisateur* (i.e. for the spat.spat∼ rendering object). spat.oper is a “perceptual operator” which provides high-level control over the acoustical quality of the virtual room [4, 5]. It allows to navigate along different perceptual axes called “perceptual factors”. These axes are mutually independent and they correlate with physical criteria observed in real concert halls. Amongst the nine perceptual factors, three are related to the source itself (presence, warmth, brillance), three characterize the room itself (reverberance, heaviness and liveness), and three account for the interaction between the source and the room (room presence, running reverberance and envelopment). spat.oper can be set to imitate the acoustics of an existing room and allows to interpolate naturally (and continuously) between different acoustic qualities.



Perceptual Factors

Reverberance	33
Heaviness	25
Liveness	35

Room Response

Room size	2000 m^3
Early min	22.22 ms
Early max	39.71 ms
Early distr	0.50
Early shape	0.50
Cluster min	42.47 ms
Cluster max	105.64 ms
Cluster distr	0.50
Reverb start	95.37 ms
Modal density	0.86

Options

✓ Active

☐ Infinite

✓ Air Absorption

Air rolloff frequency 10000.00 Hz

Crossover frequencies
Low 250.00 Hz
High 8000.00 Hz

Figure 7. Screenshot of *spat.oper*: the reverb tab.

Besides the perceptual factors, spat.oper allows to control the filtering parameters (filtering of the direct and reverberant sounds), the radiation of the sources (aperture and orientation of the radiation pattern), the characteristics of the distance attenuation law, as well as miscellaneous options. spat.oper displays as a tabbed component: there is one tab for each source (Figure 8) and one tab for each virtual room (Figure 7). Finally, spat.oper embeds a spat.viewer component for a schematic view of the sound scene.

Many other graphical user interfaces have been developed (and are progressively enhanced) for controlling, monitoring and authoring sound spatialization. It is beyond the scope of this paper to describe these interfaces in detail. A quick overview is presented in Figure 9.

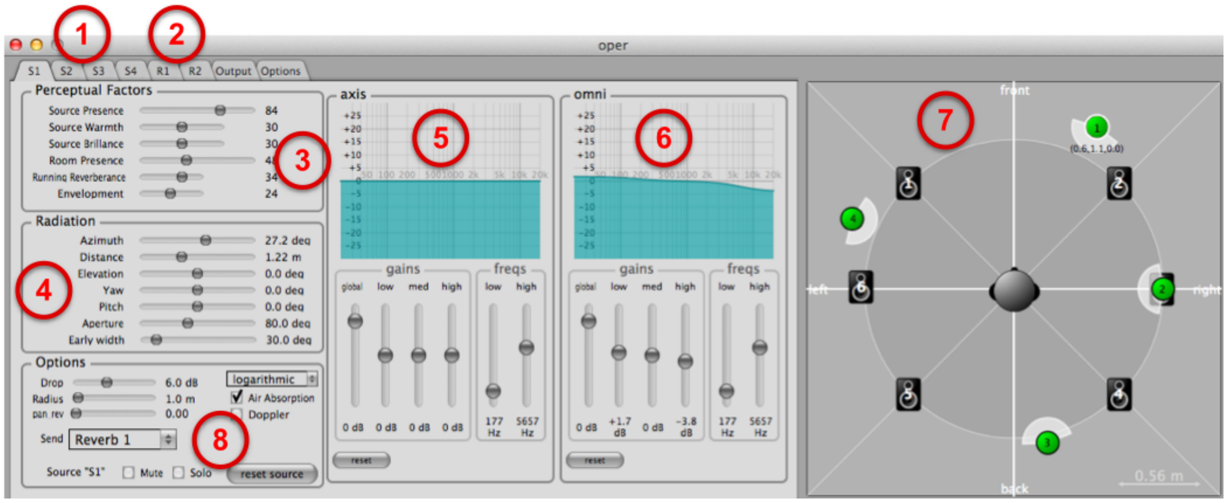
6. ROOM IMPULSE RESPONSES

Due to the increase in available processing power, convolution-based reverberation processing became widely applied during the last few decades. Several new objects have then been added to the *Spatialisateur* suite for manipulating and processing room impulse responses (IR).

6.1 (Parametric) convolution

spat.conv∼ is a multichannel convolution engine based on overlap-saved block-partitioned FFT algorithms [20]. The computation of the high latency blocks is handled in a background thread in order to take advantage of modern multi-core processors. The user can adjust the minimum block size which offers a tradeoff between audio latency (which can be set to zero) and computational load.

spat.converb∼ is another convolution processor based on the same DSP algorithms. spat.converb∼ further truncates the impulse response in four temporal segments (with adjustable



Perceptual Factors

Source Presence	84
Source Warmth	30
Source Brillance	30
Room Presence	48
Running Reverberance	34
Envelopment	24

Radiation

Azimuth	27.2 deg
Distance	1.22 m
Elevation	0.0 deg
Yaw	0.0 deg
Pitch	0.0 deg
Aperture	80.0 deg
Early width	30.0 deg

Options

Drop	6.0 dB
Radius	1.0 m
pan rev	0.00

logarithmic
[✓ Air Absorption]
Doppler
Send Reverb 1
Source "S1" Mute Solo reset source

Figure 8. *spat.oper*: source tab. ① selection of the source tab. ② selection of the room tab. ③ perceptual factors for the source. ④ localization and radiation panel. ⑤ direct sound filter. ⑥ reverberated sound filter. ⑦ visualization of the sound scene. ⑧ other options.

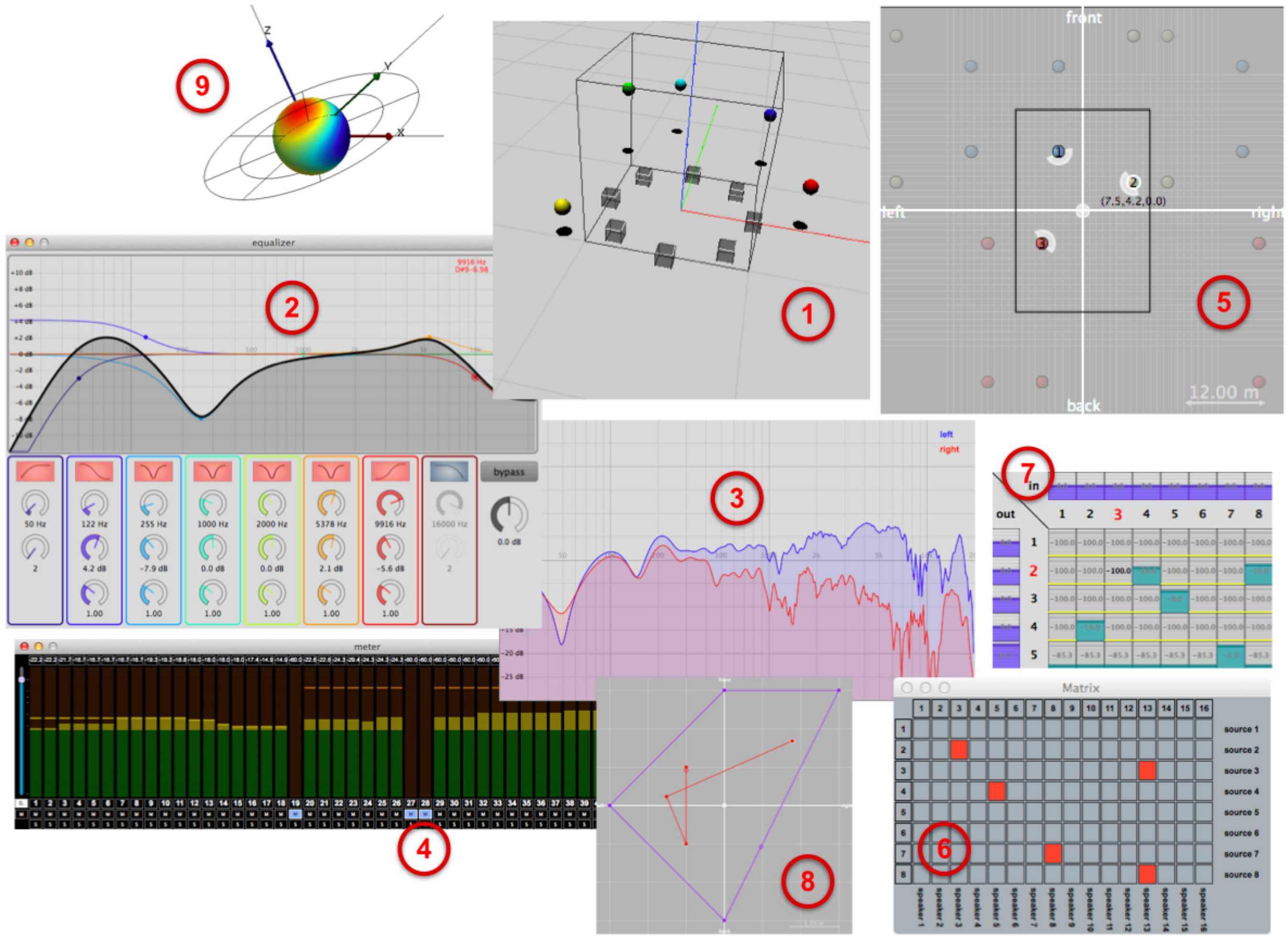


Figure 9. Several user interfaces included in Spat∼. ① *Jitter* interface for *spat.viewer*. ② parametric equalizer. ③ HRTFs inspection (magnitude/phase responses). ④ multichannel level meter. ⑤ geometrical model with image sources. ⑥ binary matrix. ⑦ matrix control. ⑧ authoring of spatial trajectories. ⑨ 3D sound field visualization.

lengths) according to the *Spatialisateur* paradigm (Figure 2); additionally a parametric filter is applied to each segment of the IR. The user can control the four filters either at the low-level (gains and cutoff frequencies) or by means of the high-level perceptual approach of spat.oper [21].

6.2 Measurement of room impulse responses

spat.smk∼ is a tool for measuring the impulse response of linear time-invariant systems using the swept-sine technique [22]. After configuration of the sweep signal (duration, waveform, frequency range, etc.), the object performs the measurement of the system (which can be multichannel), deconvolves the raw signals, and saves the data as well as meaningful metadata to disk. After each measurement, the object estimates various criteria in realtime (signal-to-noise ratio, harmonic distortion ratio, reverberation time, etc.) for controlling the quality of the measure.

In addition to that, spat.smk∼ can be linked to other acoustical tools such as spat.edc which computes and displays the energy decay curve of the IR, spat.mixingtime which estimates and displays the mixing time, or spat.ir.infos which derives many acoustical criteria (clarity, central time, early decay time, etc.).

6.3 Analysis of room impulse responses

Measured room impulse responses can be further analyzed thanks to the spat.rat object (Room Acoustics Toolbox). This external implements a time-frequency analysis of the energy decay relief (spectral energy density after any time) [23]. From this analysis it is possible to extract all the (frequency dependent) characteristics of the decay: noise level, reverberation time and initial power spectrum.

The benefits of such estimation are twofold: 1) it is possible to denoise the measured IRs (the part corrupted by noise is subtracted and replaced by a synthetic late reverberation conform to the original energy decay relief) [23]; such denoising is crucial if the IR is to be used in a convolution processor. 2) it is also possible to estimate the perceptual factors corresponding to the IR; in other words the spat.rat object can derive a set of parameters for spat.oper such that the Spat∼ FDN seamlessly mimics the measured IR.

6.4 Hybrid reverberation processor

Convolution-based approaches (such as spat.conv∼, see paragraph 6.1) guarantee for an authentic and natural listening experience as they preserve the acoustical signature of existing rooms. However convolution processors are often difficult to parametrize: the control over the reverberation effect is, in general, limited to only a few low-level parameters such as early-to-reverb ratio. Also the computational cost of such engines depends on the length of the processed IR. On the other hand, parametric reverberators (such as FDNs) are scalable and they allow for a continuous tuning of the time and frequency behavior of the room response. A commonly reported drawback of FDN rendering is the lack of authenticity in the early part of the room response (coloration artifacts, insufficient echo and/or modal densities).

spat.hybrid∼ is a new realtime audio signal processing unit that proposes a hybrid approach [24]: it combines a convolution reverb for recreating the early reflections of a measured impulse response with a feedback delay network for synthesizing the reverberation tail. The FDN is automatically adjusted so as to match the energy decay profile of the measured IR, and the generated room effect is perceptually indistinguishable from a pure convolution reverb. As a result, the hybridization approach benefits both from the authenticity of the convolution technique and the flexibility of FDN-based renderer. The spat.hybrid∼ object can further be controlled with a high-level model [24, 21].

7. MISCELLANEOUS DEVELOPMENTS

7.1 Multichannel tools

When working with massively multichannel data, users often face inappropriate or inefficient tools, even for basic operations. The *Spatialisateur* library has thus been supplemented by several simple tools for dealing with these common tasks. It is beyond the scope of this paper to examine them exhaustively and we just mention here a few examples:

- spat.sfplay∼ and spat.sfrecord∼ serve as a replacement to conventional Max/MSP sfplay∼ and sfrecord∼ objects. They exhibit similar functionalities and messaging, but they can handle up to 250 channels [8] and they support WAV RF64 format which allows the usual 4GB file size limit to be overridden.
- spat.send∼ and spat.receive∼ can be used to create a multichannel bus without patch cords, similarly to send∼ and receive∼.
- spat.virtualspeakers∼ relies on a virtual speakers paradigm [25] to down-mix any multichannel stream into a binaural stereo track preserving the spatial image of the original sound scene.
- spat.align∼ performs delay and level alignment of an arbitrary set of loudspeakers, according to their geometrical arrangement.
- Similarly, spat.delaycalibration∼ and spat.gaincalibration∼ can be used to quickly and automatically calibrate a loudspeaker setup and adjust the respective gain and delay of each channel. Unlike spat.align∼ these objects rely on in-situ acoustical measurements.

7.2 Multichannel audio effects

It is common practice to apply various audio effects to the output channels of a reproduction setup. Using standard Max/MSP objects in this context appears to be quite impracticable as they (generally) process monophonic signals only. The *Spatialisateur* library thus offers multichannel versions of a wide

[8] This corresponds to the current limitation of Max/MSP.

range of classic audio effects: spat.compressor∼ (compressor/expander), spat.eq∼ (parametric equalizer based on biquadratic filters), spat.graphiceq∼ (graphic equalizer with adjustable number of bands), spat.limiter∼, spat.noisegate∼, spat.softclipping∼, spat.dcfilter∼, etc.

7.3 SOFA support

The *Spatialisateur* package includes a large set of measured HRTFs which can be used in the binaural synthesis modules. Originally HRTF data were stored in Max/MSP "coll" files. Such textual format was not very flexible and it was not possible to embed metadata. Later, another storage format —based on SDIF [9] — has been developed. Although fully functional, this proprietary format was a strong restriction regarding the interchange with other institutions or third party applications.

In the latest versions of *Spatialisateur*, these formats (coll and SDIF) are deprecated and superseded by the "Spatially Oriented Format for Acoustics" (SOFA). SOFA [26] is a new format for storage and exchange of acoustical data, notably HRTFs. SOFA format is open, extensible, self-described, machine-independent, network-transparent, it supports data compression, and it was recently approved by the AES committee as "AES69-2015 standard for file exchange – Spatial acoustic data file format". Since 2013 Spat∼ uses *libsofa* [10] , a C++ implementation of the SOFA specifications which allows HRTFs to be loaded from the many available databases [11] .

8. CONCLUSIONS

This paper presented the latest developments of Ircam *Spatialisateur* since version 4.x, which constituted a complete refactoring. Substantial improvements and new functionalities have been introduced in every module of the software: the reverberation module, panning module and control interfaces. All in all, more than 150 external objects are shipped with the current public release [12] .

Recent research has focused on the analysis, transformation and re-synthesis of room impulses responses, with an emphasis on directional room impulses responses measured with spherical microphone arrays. Such directional IRs capture rich space-time-frequency features and they are prone to efficient manipulations in the HOA domain. Future work will consider the potential exploitation of directional room impulse responses using a *Spatialisateur*-like approach. To that purpose, the FDN processing architecture might be substituted with a convolutional or hybrid framework using an HOA-like underlying space-time representation format, and further studies on spatial perception of sound fields is needed to upgrade the perceptual factors model accordingly.

[9] Sound Description Interchange Format
[10] `http://sourceforge.net/projects/sofacoustics/`
[11] `http://hrtf.ircam.fr`
[12] `http://forumnet.ircam.fr/product/spat/`

9. ACKNOWLEDGEMENTS

The authors would like to thank everyone —researchers, composers, sound artists, sound engineers, etc.— who closely or remotely contributed and still contributes to improving Spat∼ with invaluable ideas and discussions.

10. REFERENCES

[1] J.-M. Jot and A. Chaigne, "Digital Delay Networks for Designing Artificial Reverberators," in *Proceedings of the 90th Convention of the Audio Engineering Society*, Paris, 1991.

[2] J.-M. Jot, "Real-time spatial processing of sounds for music, multimedia and interactive human-computer interfaces," *ACM Multimedia Systems Journal (Special issue on Audio and Multimedia)*, vol. 7, no. 1, pp. 55 – 69, 1997.

[3] J.-M. Jot and O. Warusfel, "A Real-Time Spatial Sound Processor for Music and Virtual Reality Applications," in *Proceedings of the International Computer Music Conference*, Banff, 1995.

[4] J.-P. Jullien, "Structured model for the representation and the control of room acoustic quality," in *Proceedings of the 15th International Congress on Acoustics (ICA)*, Trondheim, 1995, pp. 517 — 520.

[5] E. Kahle and J.-P. Jullien, "Subjective Listening Tests in Concert Halls: Methodology and Results," in *Proceedings of the 15th International Congress on Acoustics (ICA)*, Trondheim, June 1995, pp. 521 – 524.

[6] J. Bresson, C. Agon, and G. Assayag, "OpenMusic. Visual Programming Environment for Music Composition, Analysis and Research," in *Proceedings of ACM MultiMedia (OpenSource Software Competition)*, Scottsdale, 2011.

[7] J.-M. Jot, V. Larcher, and O. Warusfel, "Digital Signal Processing Issues in the Context of Binaural and Transaural Stereophony," in *Proceedings of the 98th Convention of the Audio Engineering Society*, Paris, Feb. 1995.

[8] V. Pulkki, "Virtual Sound Source Positioning Using Vector Base Amplitude Panning," *Journal of the Audio Engineering Society*, vol. 45, no. 6, pp. 456–466, June 1997.

[9] J.-M. Jot, V. Larcher, and J.-M. Pernaux, "A comparative study of 3-D audio encoding and rendering techniques," in *Proceedings of the 16th Audio Engineering Society International Conference on Spatial Sound Reproduction*, Rovaniemi, 1999.

[10] T. Lossius, P. Balthazar, and T. de la Hogue, "DBAP - Distance-Based Amplitude Panning," in *Proceedings of the International Computer Music Conference*, Montréal, 2009.

[11] R. Sadek and C. Kyriakakis, “A Novel Multichannel Panning Method for Standard and Arbitrary Loudspeaker Configurations,” in *Proceedings of the 117th Audio Engineering Society Convention*, San Francisco, 2004.

[12] J. Daniel, “Représentation de champs acoustiques, application à la transmission et à la reproduction de scènes sonores complexes dans un contexte multimédia,” Ph.D. dissertation, Université de Paris VI, 2001.

[13] M. Noisternig, T. Carpentier, and O. Warusfel, “Espro 2.0 – Implementation of a surrounding 350-loudspeaker array for sound field reproduction.” in *Proceedings of the 25th Audio Engineering Society UK Conference*, 2012.

[14] J. Daniel and S. Moreau, “Further Study of Sound Field Coding with Higher Order Ambisonics,” in *Proceedings of the 116th Audio Engineering Society Convention*, Berlin, 2004.

[15] F. Zotter, H. Pomberger, and M. Noisternig, “Energy-Preserving Ambisonic Decoding,” *Acta Acustica united with Acustica*, vol. 98, 2012.

[16] M. Pollow, K.-V. Nguyen, O. Warusfel, T. Carpentier, M. Müller-Trapet, M. Vorländer, and M. Noisternig, “Calculation of Head-Related Transfer Functions for Arbitrary Field Points Using Spherical Harmonics Decomposition,” *Acta Acustica united with Acustica*, vol. 98, 2012.

[17] R. O. Duda and W. L. Martens, “Range dependence of the response of a spherical head model,” *Journal of the Acoustical Society of America*, vol. 104, no. 5, pp. 3048–3058, 1998.

[18] D. Romblom and B. Cook, “Near-Field Compensation for HRTF Processing,” in *Proceedings of the 125th Audio Engineering Society Convention*, San Francisco, 2008.

[19] A. Baskind, T. Carpentier, M. Noisternig, O. Warusfel, and J.-M. Lyzwa, “Binaural and transaural spatialization techniques in multichannel 5.1 production,” in *Proceedings of the 27th Tonmeistertagung – VDT International Convention*, Köln, November 2012.

[20] W. G. Gardner, “Efficient Convolution without Input-Output Delay,” in *Proceedings of the 97th Convention of the Audio Engineering Society*, San Francisco, 1994.

[21] M. Noisternig, T. Carpentier, and O. Warusfel, “Perceptual Control of Convolution Based Room Simulators,” in *The Acoustics 2012 Hong Kong Conference*, Hong Kong, 2012.

[22] A. Farina, “Simultaneous measurement of impulse response and distortion with a swept-sine technique,” in *Proceedings of the 108th Convention of the Audio Engineering Society*, Paris, 2000.

[23] J.-M. Jot, L. Cerveau, and O. Warusfel, “Analysis and Synthesis of Room Reverberation Based on a Statistical Time-Frequency Model,” in *Proceedings of the 103rd Convention of the Audio Engineering Society*, New York, 1997.

[24] T. Carpentier, M. Noisternig, and O. Warusfel, “Hybrid Reverberation Processor with Perceptual Control,” in *Proceedings of the 17th Int. Conference on Digital Audio Effects (DAFx-14)*, Erlangen, Sept. 2014.

[25] H. Møller, “Fundamentals of Binaural Technology,” *Applied Acoustics*, vol. 36, pp. 171 – 218, 1992.

[26] P. Majdak, Y. Iwaya, T. Carpentier, R. Nicol, M. Parmentier, A. Roginska, Y. Suzuki, K. Watanabe, H. Wierstorf, H. Ziegelwanger, and M. Noisternig, “Spatially Oriented Format for Acoustics: A Data Exchange Format Representing Head-Related Transfer Functions,” in *Proceedings of the 134rd Convention of the Audio Engineering Society*, Roma, May 2013.