Grant Agreement No.: 687645
Research and Innovation action
Call Topic: H2020 ICT-19-2015

Object-based broadcasting – for European leadership in next generation audio experiences

D3.3: Object-based capture

Version: v1.2

Deliverable type	R (Document, report)
Dissemination level	PU (Public)
Due date	31/05/17
Submission date	10/07/17
Lead editor	Nicolas Epain (b<>com)
Authors	Nicolas Epain (b<>com), Emanuel Habets (FhG), Markus Noisternig (IRCAM)
Reviewers	Werner Bleisteiner (BR)
Work package, Task	WP3, T3.1
Keywords	Recording; 3D audio; Microphone array; Sound scene analysis; Reverberation

Abstract

This deliverable gives an overview of the work done within Task T3.1 (Capturing) of the ORPHEUS project regarding object-based capture. The constraints and requirements specific to the context of object-based audio content production are described. The different tools and methods designed by the partners are presented and their use is discussed in regard to those constraints and requirements.

[End of abstract]

Document revision history

Version	Date	Description of change	List of contributor(s)
v0.1	01/06/2017	Initial (empty) version.	N. Epain
v0.2	19/06/2017	Skeleton, abstract, etc.	N. Epain
v0.3	22/06/2017	Added FhG contributions	E. Habets
v0.4	26/06/2017	Added most of b<>com's contributions	N. Epain
v0.5	27/06/2017	Added IRCAM's contribution	M. Noisternig
V1.0	28/06/2017	Report ready for review	N. Epain
V1.1	29/06/2017	Reviewed version	W. Bleisteiner
V1.2	10/07/17	Final editing after PMC approval and submission	U. Herzog

Disclaimer

This report contains material which is the copyright of certain ORPHEUS Consortium Parties and may not be reproduced or copied without permission.

All ORPHEUS Consortium Parties have agreed to publication of this report, the content of which is licensed under a Creative Commons Attribution-NonCommercial-NoDerivs 3.0 Unported License[1].

Neither the ORPHEUS Consortium Parties nor the European Commission warrant that the information contained in the Deliverable is capable of use, or that use of the information is free from risk, and accept no liability for loss or damage suffered by any person using the information.

[1] http://creativecommons.org/licenses/by-nc-nd/3.0/deed.en_US

Executive Summary

This deliverable presents the work achieved within the Capturing task of the ORPHEUS project. Recording audio signals is the very first step in the creation of radio programs and sound engineers have been using microphones for more than a century. However, the adoption of an object-based paradigm in audio content production induces the need for new recording devices and techniques that facilitate this task.

The different methods investigated by the partners are described and their use is discussed in the context of object-based audio content production. These methods can be divided in two main categories. The first category of methods aims at capturing 3D audio scenes. As an example, the partners developed efficient software tools that convert the signals recorded my microphone arrays into multichannel or Higher-Order Ambisonic sound scenes. The second category aims at capturing audio object metadata, such as object positions or reverberation parameters.

As the aim of this work was to facilitate the work of sound engineers, most of the novel audio capture methods have been implemented as VST plugins, which are compatible with most Digital Audio Workstations (DAWs) on the market and in particular with Sequoia, the DAW used in the ORPHEUS project.

Table of Contents

List of Figures

List of Tables

Abbreviations

HOA	Higher Order Ambisonics
SMA	Spherical Microphone Array
CMA	Circular Microphone Array
FDN	Feedback Delay Network
VST	Virtual Studio Technology
IBC	International Broadcasting Convention
DOA	Direction of Arrival
STFT	Short-Term Fourier Transform
ADM	Audio Definition Model
SNR	Signal-to-Noise Ratio
RIR	Room Impulse Response
DRIR	Directional Room Impulse Response
HRTF	Head-Related Transfer Function

1 Introduction

Capturing audio signals with microphones constitutes the very beginning of the audio broadcasting chain. For decades, sound engineers have been perfecting recording and mixing techniques that are specifically suited to channel-based audio formats. In order to capture audio objects, new tools and algorithms must be designed. Ideally, these tools should be reliable and intuitive enough that sound professionals have no major difficulty adapting to the new audio production processes. Designing such tools and methods is the aim of Task T3.1, “Capturing”, in the ORPHEUS project. This report gives an overview of the work achieved within this task.

To begin, the title of this deliverable, “object-based capture”, requires clarification. In the ORPHEUS project, object-based must be understood in a broad sense and the word object can refer to various entities, from a series of audio grains that constitute a program together to multiple audio tracks containing reverberation. In this context, the meaning of this deliverable’s title is “capture for object-based audio production”. In the object-based audio workflow, content more and more often consists of a mixture of signals using three different types of sound scene representation:

- Channel-based content: typically two or more audio signals to be played by loudspeakers located in specific positions in space. Stereo or 5.1 audio tracks are both examples of channel-based content.
- What is referred to as “object-based” audio: An audio object is typically a mono or stereo signal with associated metadata describing its position in space, over time, in width and diffuseness, etc.
- Scene-based content: whereby the signals represent an entire sound scene but, opposite to channel-based content, individual tracks do not correspond to individual speakers but rather to “modes” used to decompose the scene. Higher-Order Ambisonics (HOA), in which sound scenes are decomposed over a basis of spherical harmonic functions, is a good example of scene-based audio format.

Modern object-based audio formats, including MPEG-H and ADM, support all three types of sound scene representation. Accordingly, all three have been addressed by the ORPHEUS partners in their contribution to the capturing task. Multi-channel or scene-based audio can be captured with the help of 2D or 3D microphone arrays, which make it possible to retain the spatial characteristics of the recorded sound field. Contributions in this area are presented in Section 2.

Regarding the capture of sound “objects”, understood as isolated sound sources, there has been a considerable amount of research on sound source separation[2] over the past few decades. However, the quality of the signals extracted by the means of source separation algorithms is not suitable for use in production. Additionally, source separation techniques typically require an important amount of computational power, which make them impractical for sound engineers. Hence our approach has been to consider that sound objects would keep being captured using proximity microphones[3] in the near future. In this context, the question remains as to how to extract the metadata corresponding to the signals recorded by proximity microphones. Contributions in regards to this question are presented in Section 3.

One of the problems that has been slowing down the adoption of scene-based formats (in particular HOA) in production is the lack of tools that are routinely used for mixing channel-based audio. In order to fill this gap, the partners designed tools allowing to monitor and perform basic editing

[2] The idea of sound source separation algorithms is to decompose microphone signals, typically recorded using a microphone array, either into monophonic signals corresponding to the sounds emitted by the different sources, or into multichannel signals corresponding to the contribution of the different sound sources to the scene, including reverberation.

[3] Proximity microphones are microphones used in the vicinity of the sound sources, for instance the mouth of a singer.

operations in the HOA format. These tools are reviewed in Section 4.

Lastly, reverberation is a crucial aspect of audio production both for musical recordings and for radio drama and documentary. Object-based audio production induces new needs in regard to reverberation. In order to create realistic immersive content, reverberation must be either recorded, for instance with a microphone array, or synthesised using a high-quality reverb engine. In the latter case, an interesting problem is how to set the parameters of the reverb engine so that the reverberation characteristics remain as close as possible to that of a specific room, but become more adaptive to the reproduction environment and situation. Section 5 reviews the contributions of the partners in regard to these questions.

2 Capture of 3D audio scenes

In this section we review the contributions of the partners in regard to recording 3D audio scenes using microphone arrays.

2.1 3D Audio Capture with a horizontal circular array

Current solutions for capturing high-quality 3D audio require relatively large microphone setups, which can be impractical in production scenarios. Our goal was to develop a solution that does not require such a large microphone setup and can easily be integrated in a mobile or handheld device. To accomplish this goal, a 3D audio capturing algorithm was developed to transform microphone signals into an output format that is suitable for 3D audio reproduction. Typical output formats are stereo, 5.1, or 7.1+4 loudspeaker signals. The algorithm is designed to support practical microphone setups of three or more microphones, which can be arranged in an almost arbitrary configuration to achieve high flexibility in practice. An example microphone configuration is shown in Figure 1, which consists of 8 microphones forming a horizontal circular array. Even though the microphone array is planar (i.e., two-dimensional), it allows 3D audio capturing and reproduction.

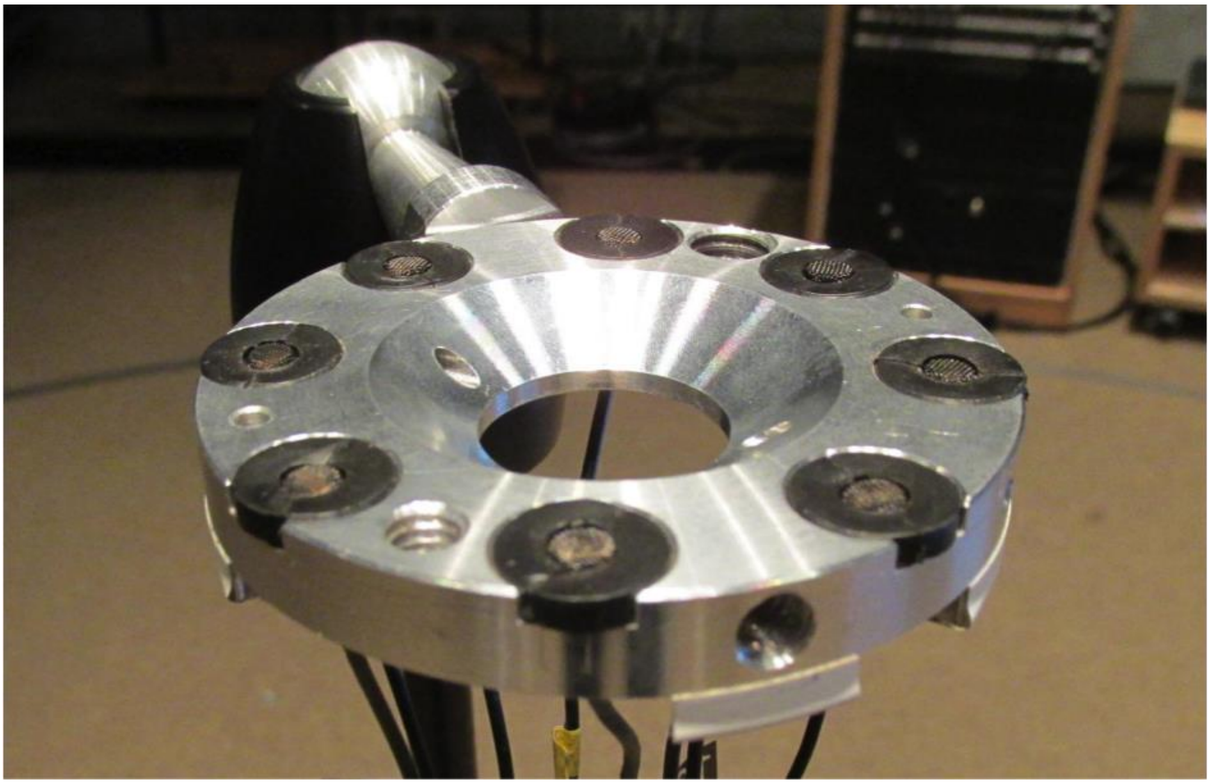


Figure 1: Example of a cylindrical microphone array

The 3D audio capturing algorithm adopts a perceptually motivated approach. This means that the output signals are generated such that during playback the cues that are relevant for the human perception of spatial sound are recreated at the listening position. The algorithm provides a high spatial resolution in situations where humans can localize sounds very well, and a diffuse rendering in situations where humans perceive the sound from all directions. Therefore, an accurate reproduction of both directional sounds and ambiance is achieved.

To achieve the desired flexibility with respect to the output formats and microphone setup, the 3D audio capturing algorithm is based on a parametric description of the sound field. The sound field is described in the time-frequency domain by means of a reference audio signal and parametric side information, namely the direction-of-arrival (DOA) of the sound and the so-called diffuseness. The latter parameter describes how diffuse the sound field is. The underlying parametric spatial sound processing is described in detail in [1].

The parametric spatial sound processing consists of the two processing blocks depicted in Figure 2:

- Signal analysis: The signal analysis block takes the microphone signals as input, transforms them into the time-frequency domain using a filter bank, and estimates the parametric description of the spatial sound. This includes estimating the DOA for each time and frequency, the diffuseness, and the reference audio signal containing the spectrum of the sound.
- Signal synthesis: The signal synthesis block uses the parametric description of the spatial sound generated in the analysis block to synthesize the playback signals of the desired output format. For this purpose, the reference audio signal first is decomposed into a direct sound and diffuse sound component based on the diffuseness parameter. Both sound components are then used to synthesise the loudspeaker signals, which are finally transformed back into the time domain using an inverse filter bank. The loudspeaker signal synthesis exploits the DOA information, namely to reproduce the direct sound component from the original direction.

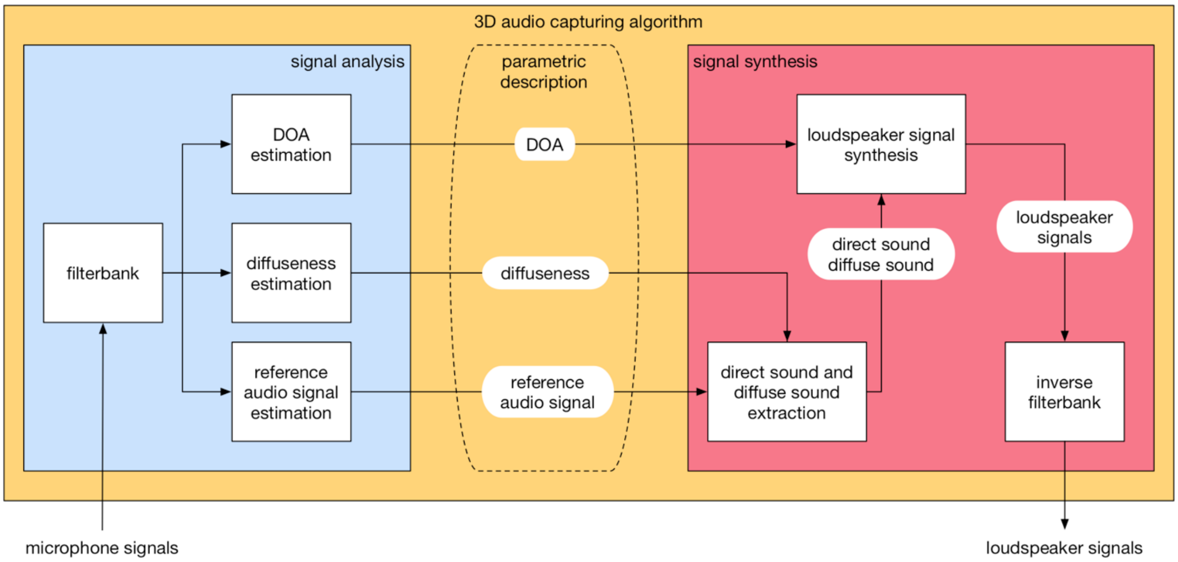
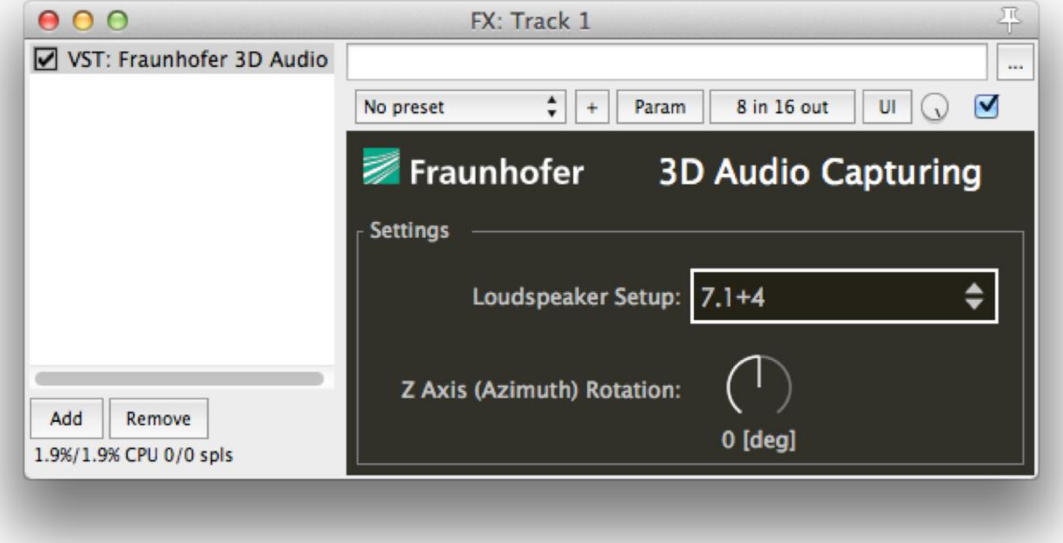


Figure 2: Block diagram of the 3D audio capturing algorithm

During this project, the 3D audio capturing algorithm was integrated into a real-time VST plugin, depicted in Figure 3. The plugin supports the circular microphone array shown in Figure 1. Moreover, various 2D or 3D loudspeaker output formats are supported, such as 5.1 and 7.1+4, which can be selected by the user. Note that thanks to the parametric description of the spatial sound, it is possible to easily apply transformations to the synthesized sound scene, such as sound scene rotations. In fact, this can be achieved by modifying the parameters of the parametric sound field description, e.g., the DOA parameter, before carrying out the signal synthesis. This is useful for example to align the acoustical image to a visual image of a camera when combining the audio recording with a video recording. The rotation feature was implemented in the VST plug-in and it enables sound scene rotation around the z-axis of the coordinate system.



Loudspeaker Setup: 7.1+4

Z Axis (Azimuth) Rotation: 0 [deg]

Figure 3: 3D audio capturing VST plug-in

2.2 3D Audio Capture with spherical microphone arrays

HOA sound scenes are typically recorded using spherical microphone arrays (SMA), that is, microphone arrays comprised of a number of microphone capsules distributed inside or over the surface of a sphere. SMAs can either be “rigid”, *i.e.* the capsules are flush-mounted on a solid spherical baffle, or “open”, *i.e.* the microphone capsules are simply located at an equal distance from the centre of the array. Some SMAs are available for purchase (for instance, MH Acoustics’ Eigenmike[4] or Soundfield’s SPS200) and, with the increasing interest in immersive media such as 360 videos they are more and more often used in production.

In general, SMA manufacturers provide the hardware or software allowing to convert the signals recorded using microphone arrays to a format in which they can be used more easily (typically ambisonic or HOA). However, it is not always clear exactly how this processing is done and the quality of the conversion can be relatively poor. This may be one of the reasons why SMAs have had the reputation of producing signals with relatively low fidelity, and especially mediocre timbre. In order to facilitate the adoption of SMAs for the production of immersive audio content, b<>com investigated how to process SMA-recorded signals so as to provide HOA signals with the highest quality possible.

From a general standpoint, HOA signals are obtained by applying what is often referred to as “HOA-encoding” filters to the microphone signals. In the frequency domain, this operation can be expressed by the following matrix-vector product:

$$\mathbf{b}(k) = \mathbf{E}(k)\ \mathbf{s}(k),$$

where $\mathbf{b}(k)$ denotes the vector of the HOA signals, $\mathbf{s}(k)$ denotes the vector of the microphone signals, $\mathbf{E}(k)$ is a matrix of HOA-encoding filters and k is the wave number. In general the coefficients of matrix $\mathbf{E}(k)$ are calculated by using a theoretical model of the array’s acoustical behaviour. For example, rigid SMAs are typically modelled as perfectly omnidirectional microphones mounted on the surface of a perfectly rigid sphere. These models neglect a number of acoustical effects, such as

[4] Eigenmike is a registered trademark of MH Acoustics

diffraction by the microphone stand, or the fact that microphone capsules are generally not omnidirectional at high frequencies and these approximations result in HOA signals of lesser quality.

In order to improve the quality of the HOA signals recorded using SMAs, we characterised the acoustic characteristics of three commercially available SMAs (the Eigenmike, Sennheiser's Ambeo[5] and Embrace Cinema's Brahma) through calibration measurements. These measurements were done in a facility that is normally used for recording Head-Related Transfer Functions (HRTFs). The facility, shown in Figure 4, consists of an anechoic room equipped with loudspeakers and a turntable. The subject or microphone array is placed on the turntable, which is rotated in steps of a few degrees. For each turntable position, impulse responses are recorded for every loudspeaker. The measurement process resulted in impulse responses recorded for sound source directions located in every possible direction (with the exception of elevations below -75 degrees) with a resolution of a few degrees.

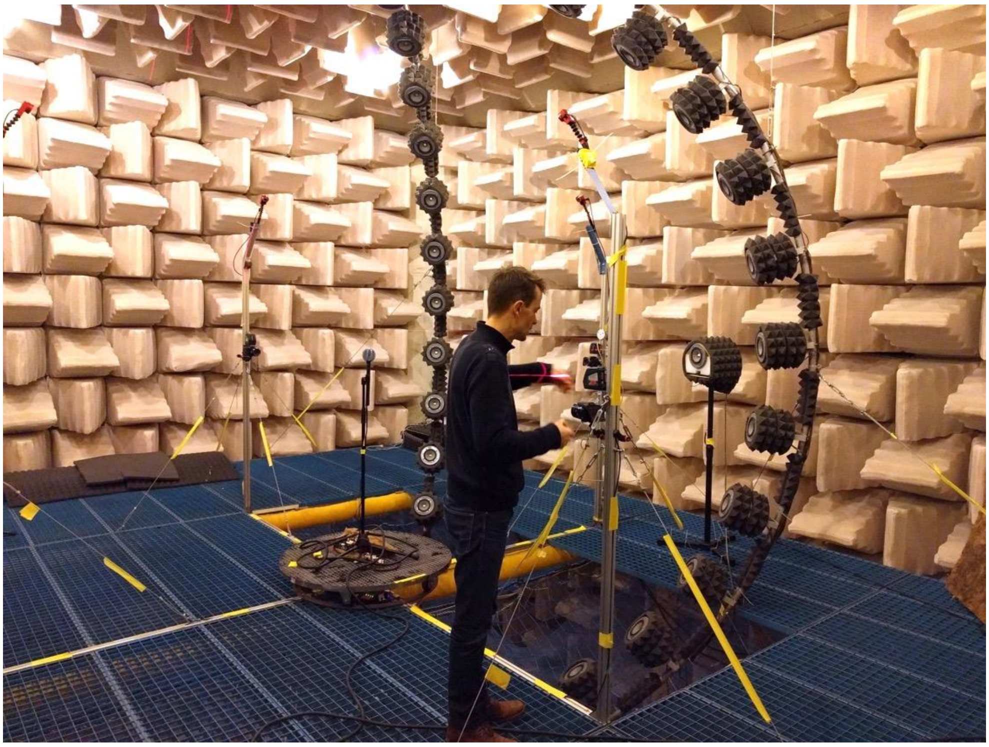


Figure 4: Setting up the acoustic characterisation of an SMA in the HRTF measurement facility.

The HOA-encoding filters were then computed from the measured transfer functions as follows. First, because the positions of the sources are known, we can form the matrix, **B**(*k*), of the HOA coefficients corresponding to the source positions:

$$\mathbf{B}(k) = \mathbf{W}(k)\,\mathbf{Y},$$

where **W**(*k*) is diagonal matrix which depends on the frequency value and source distance and **Y** is the matrix of the spherical harmonic function values for the source directions:

$$\mathbf{Y} = [\ \mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_N\ ],$$

$$\mathbf{y}_n = [\ Y_0^0(\vartheta_n, \varphi_n), Y_1^{-1}(\vartheta_n, \varphi_n), \ldots, Y_L^L(\vartheta_n, \varphi_n)\ ]^{\mathrm{T}},$$

[5] Ambeo is a registered trademark of Sennheiser

where $Y_l^m(.)$ is the order-l, degree-m real-valued spherical harmonic function and (ϑ_n, φ_n) denotes the elevation and azimuth of the *n*-th source position, respectively. In order to find the HOA-encoding filter coefficients ensuring the lowest error possible, we then must solve the following optimization problem for every frequency value:

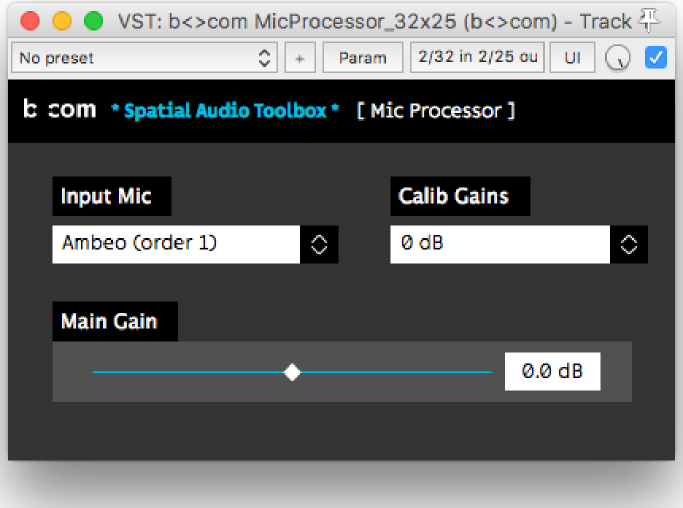
$$\text{minimize} \quad \mathbf{B}(k) - \mathbf{X}\,\mathbf{T}(k) \quad {}_2 \text{ for } \mathbf{X},$$

where $\mathbf{T}(k)$ is the matrix of the source to microphone transfer functions. The set of optimal filter coefficients are thus given by:

$$\mathbf{E}(k) = \mathbf{B}(k)\,\mathbf{T}(k)^{-1}.$$

Note that, in practice, the inversion of matrix $\mathbf{T}(k)$ is regularized so as to avoid unreasonable amplification of the microphone signals, which would result in an excessive amount of measurement noise in the HOA signals. As well, note that the encoding filters are further equalised so that the perceived timbre of the HOA signals is as flat as possible. The equalisation procedure is described in further detail in the ORPHEUS deliverable D3.5.

Additionally, a VST plugin has been created which implements the application of the HOA-encoding filters calculated using the procedure described above. A picture of the plugin's user interface is shown in Figure 5. The plugin simply proposes a selection of encoding filters, as well as gain settings.


VST: b<>com MicProcessor_32x25 (b<>com) - Track
No preset
+
Param
2/32 in 2/25 ou
UI
b com
* Spatial Audio Toolbox *
[Mic Processor]

Input Mic: Ambeo (order 1)

Calib Gains: 0 dB

Main Gain: 0.0 dB

Figure 5: User interface for b<>com's MicProcessor VST plugin.

3 Capture of object metadata

In this section, we describe the work done by b<>com regarding the extraction of audio object metadata from audio recordings. We consider a scenario where both proximity microphones (also referred to as "spot" microphones) and a microphone array are used to record sound sources such as actors or musical instruments. The aim is to estimate the position of the objects recorded by the spot microphones, relative to the microphone array.

In the future, it is likely that object-based audio capturing will be facilitated by the use of infrared or electromagnetic tracking devices. However, we believe there is merit in determining the source positions based on the audio data, because such method could be used retrospectively to process recordings done prior to using such tracking devices.

Note that this work will be elaborated in further detail and presented at the IBC in September 2017 [2].

3.1 Problem formulation

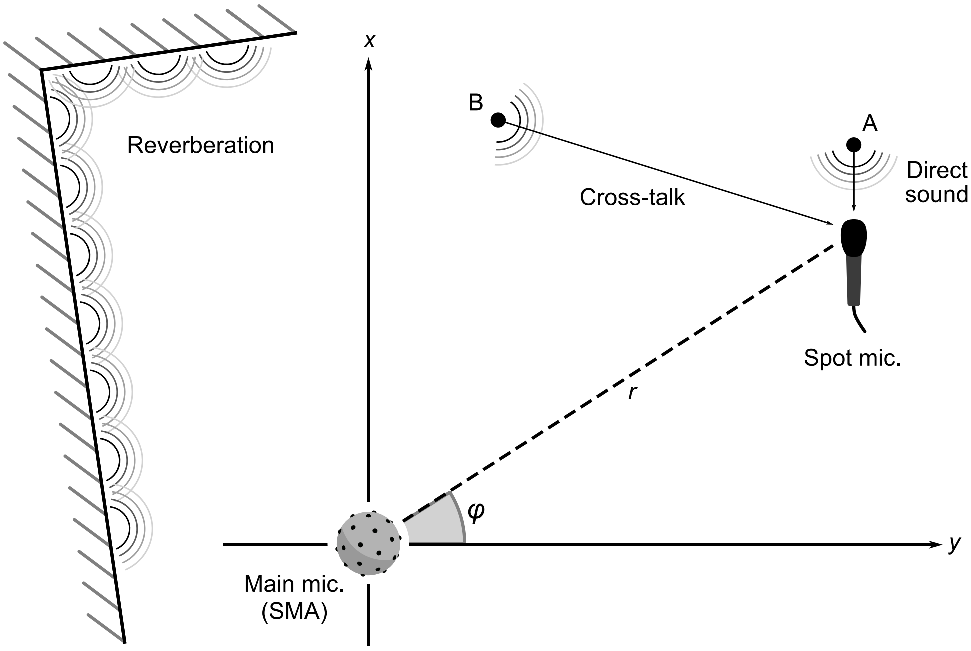


Figure 6: Problem setup: estimating the location of source A by analysing the signals recorded by the "Main" and "Spot" microphones (please refer to text for more details).

Consider the scenario described by Figure 6. A spherical microphone array (SMA) is used to record a scene comprised of several sound sources (A, B), in an environment that may include walls. At least one of the sound sources is equipped with a proximity microphone. In the following we refer to the SMA as the "main" microphone and to the proximity microphone as the "spot" microphone. The signals recorded by the main microphone are processed and converted to a spherical harmonic representation, i.e. in ambisonics' B-format or as Higher-Order Ambisonics (HOA) signals up to order L [3]. Our aim is to analyse the signals recorded by the main and spot microphones to determine the direction and distance of source A relative to the main microphone.

Note that, for simplicity, we make the hypothesis that source A is close enough to the spot microphone and that its position relative to the main microphone is approximately the same as that of the spot microphone. In other words, given that source A is an actor or singer, we assume that her/his mouth is close enough to the spot microphone that the corresponding shift in position would be negligible when heard from the main microphone location. We also neglect the presence of self-noise in the spot microphone signal.

3.2 The MainSpot algorithm

In previous work [4], b<>com has presented an algorithm to estimate the location of source A by analysing the signals recorded by an order-1 ambisonic microphone (such as Sennheiser's Ambeo) together with that recorded by a spot microphone. The algorithm, which we refer to as "MainSpot" is summarised in Figure 7. The order-1 ambisonic signals provided by the main microphone are denoted *w*(*t*), *x*(*t*), *y*(*t*) and *z*(*t*), while the signal recorded by the spot microphone is denoted *s*(*t*). In a first step the "omni" signal *w*(*t*) is cross-correlated with the spot signal *s*(*t*) to estimate a delay value. Next, this delay is applied to the spot signal *s*(*t*) so that it is in phase with the contribution of source A to the main signal. The delayed spot signal is then projected on the "main" signals x, y and z to obtain a vector, which points to the estimated source direction. Lastly, the coordinates of the direction vector are converted to elevation and azimuth values ϑ and φ.

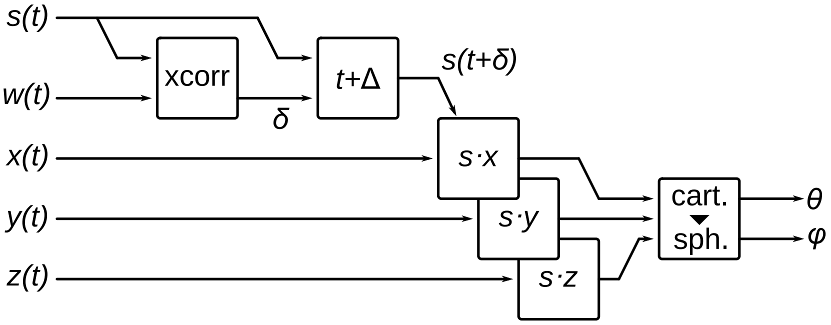


Figure 7: Flow diagram of the MainSpot algorithm.

The MainSpot algorithm has the advantage that it requires very little computational power. This allowed us to implement the algorithm in real time in the form of a VST plugin, which is shown in Figure 8. In its current form, the plugin simply illsutrates the estimated source elevation and azimuth, as well as the delay between the spot and main micrphone signals. Therefore the plugin is only assisting sound engineers, which still have to set the direction of objects by hand in the DAW.

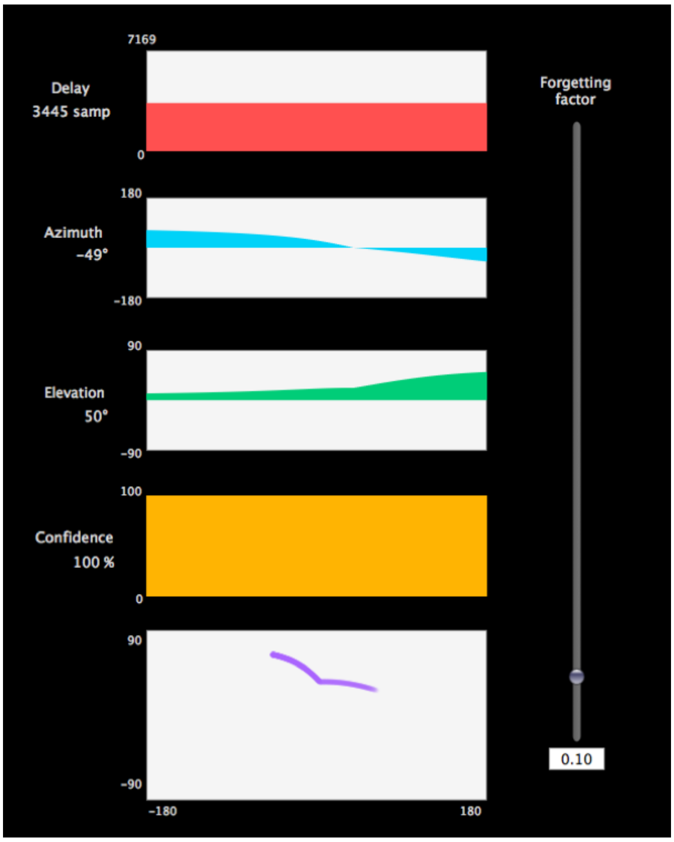


Figure 8: Graphical user interface of the MainSpot VST plugin.

However, testing over signals recorded during music and drama performances revealed that the MainSpot algorithm was not very robust in the presence of interfering sources or reverberation. We identified several causes for this issue and designed a new algorithm which improves the robustness of the estimation in actual recording conditions. One of the advantages of the novel algorithm over the MainSpot is that it is designed to work with HOA signals with no limit on the HOA order, while the previous version was limited to order 1.

In the following, we present results demonstrating the performance of this new algorithm.

3.3 Results

The performance of the novel source localisation algorithm was first tested through numerical simulations. Two series of simulations were carried out, whereby we assessed the robustness of the method to: 1) the presence of diffuse noise and 2) the presence of an interfering source. In both cases, the performance of the algorithm was found to be excellent and the error in the estimation of the source direction after convergence of the algorithm was less than a few degrees in every trial.

In order to assess the performance of our source localisation algorithm in realistic conditions, we also applied it to files recorded during an actual musical performance. The recording setup was the following. The musicians were distributed along a circle, about two meters in radius. Each sound source was recorded with a proximity microphone. In addition, a 32-channel SMA was located at the centre of the circle, approximately 2 meters above the floor. Directly in front of the SMA was located a singer, sitting. On the sides were a clarinet, cello, harp, flute and percussions.

Results of the analysis are presented in Figure 9. During the first 4 seconds of the recording, both the direction and delay estimations are unstable. This is because the singer has not started singing at this time, and the spot microphone picks up signals originating from the percussions and cello. After the singer starts singing (4 s onwards), the delay estimation becomes very stable at approximately 230 samples and remains so until the end of the recording. It takes a bit longer for the estimated source direction to converge to the expected value (0° azimuth and -30° elevation). Nevertheless, after convergence (12 s onwards), the estimated source direction remains within 5 to 10 degrees of the expected direction.

Regarding the computational cost of the algorithm, it is greater than that of the original MainSpot algorithm. Hence it is unlikely that this method will be implemented in a VST plugin. Nevertheless we will continue this work and implement the method as a standalone tool that exports the estimated positions as ADM metadata.

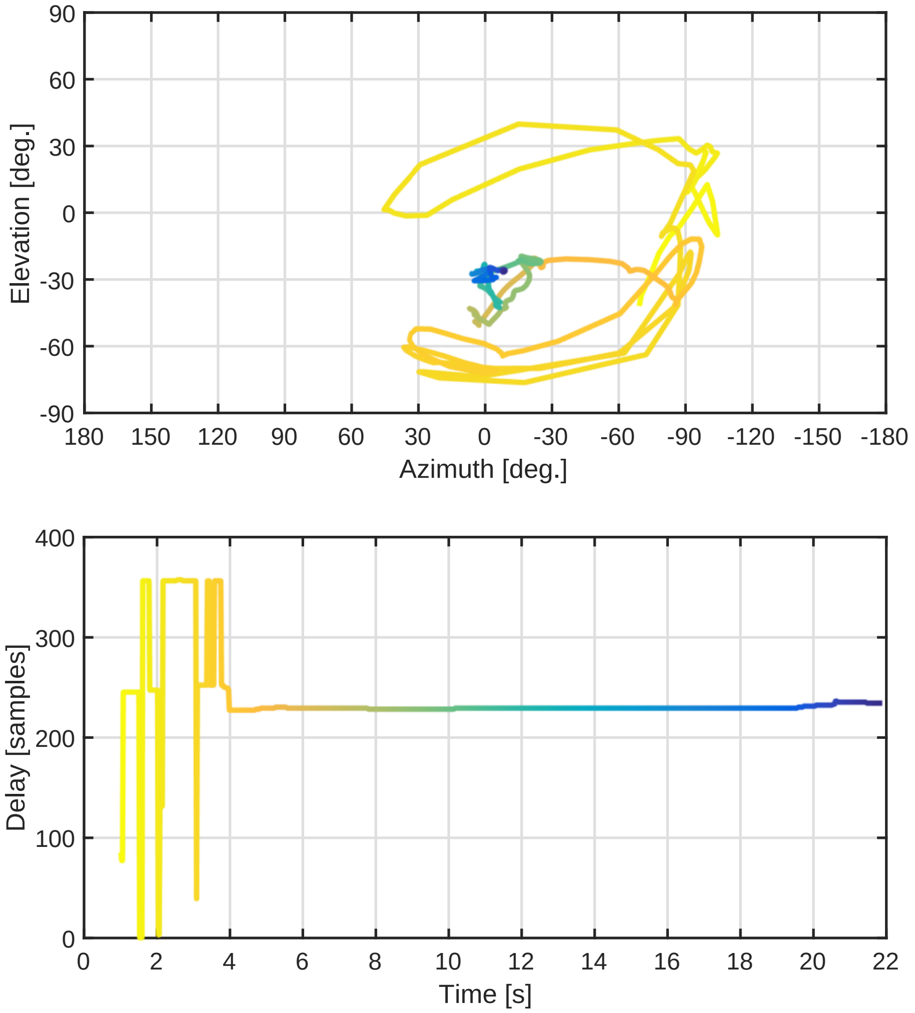


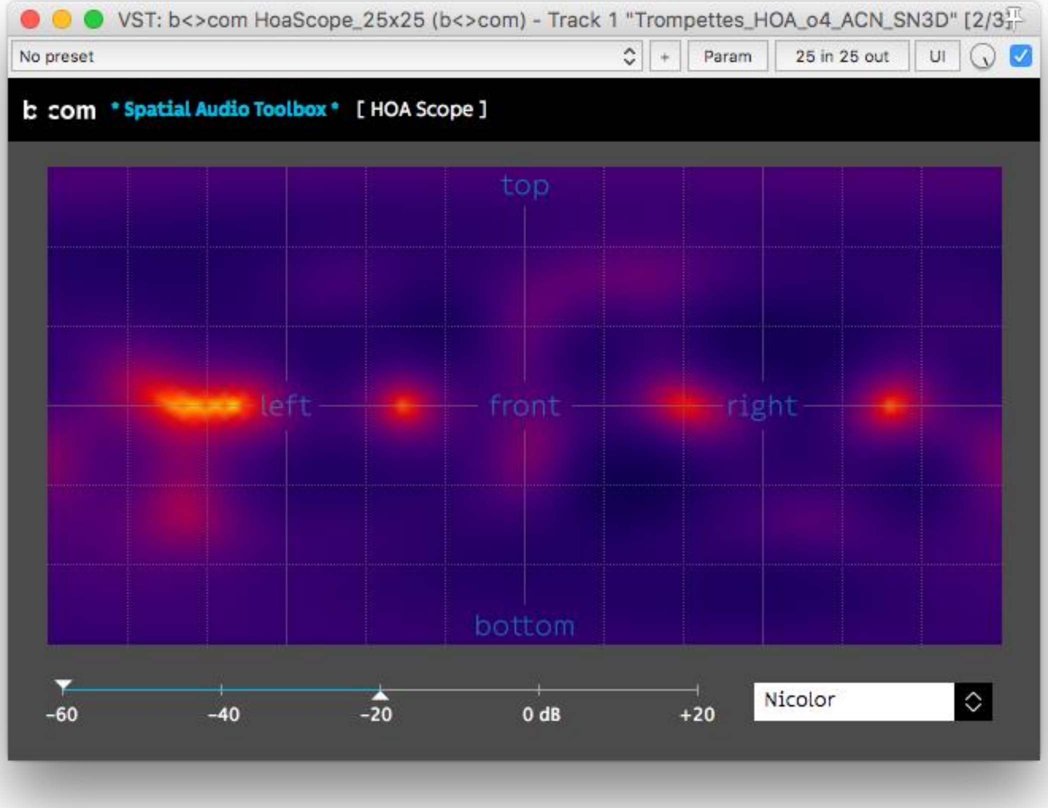
Figure 9: Trace of the source localisation algorithm's output when run on recorded signals. The colour of the line indicates at which point in the signals the position and delay were estimated.

4 Audio scene analysis and edition

In scene-based audio representations, among which is HOA, individual tracks do not correspond to specific directions of space as they do in channel-based formats. This makes basic operations such as monitoring, mixing and editing a bit more difficult and very few tools provide functionalities of this kind. In this section we present software developed by the partners to help analyse and edit HOA sound scenes.

4.1 Spatial monitoring of HOA sound scenes

When recording HOA signals it may be difficult to monitor what is going on around the microphone array. In particular, one might want to have an estimate of the sound source positions and intensities. Information of this kind can greatly facilitate the work of sound engineers, in order to set the position and orientation of the microphone array, for instance.



VST: b<>com HoaScope_25x25 (b<>com) - Track 1 "Trompettes_HOA_o4_ACN_SN3D" [2/3

No preset | Param | 25 in 25 out | UI

Figure 10: User interface for b<>com's HoaScope plugin.

In order to fill this gap, b<>com designed a VST plugin that performs a spatial analysis of HOA signals, which we refer to as the *HoaScope*. The graphical interface of the plugin is shown in Figure 10. The plugin displays a map (equirectangular representation of the sphere) of the incoming acoustic energy as a function of the direction in real time. In other words, each pixel on the map corresponds to a direction in space and the pixel colours indicate the RMS level estimated for the corresponding directions. The slider located in the lower-left part of the interface let the user choose how the colour axis is mapped to RMS values: in the example shown in Figure 10, any value below -60 dB RMS is displayed as a black pixel, while values above -20 dB RMS are displayed as white pixels.

Note that some tools providing similar functionalities exist on the market. For example, Blue Ripple

Sound's "O3A Visualiser" plugin also displays a map of the incoming acoustic energy. However, the map displayed by the *HoaScope* has a much higher resolution than that provided by the O3A Visualiser for the same HOA signals. This is because the two plugins use different algorithms to estimate the energy as a function of direction.

One remaining issue with the *HoaScope* is that it processes the input HOA signals in the time domain. While this is not a problem for synthetic HOA signals, it may induce errors in the energy calculation when applied to signals recorded with a microphone array. Indeed, the spatial resolution (order) of the HOA signals recorded using an SMA strongly depends on the geometry of the array, and in particular on its size relative to the wavelength. For example, the effective order of the HOA signals recorded with an Eigenmike is as follows:

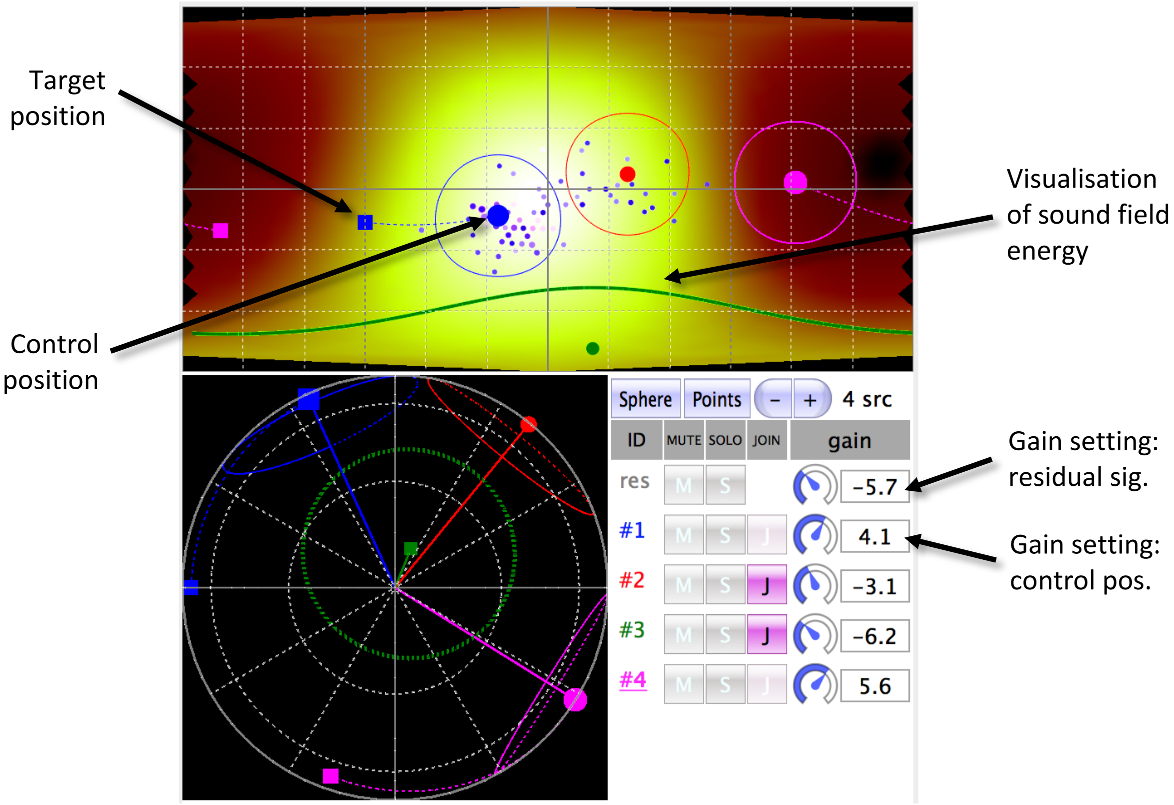
Frequency range	0-150 Hz	150-350 Hz	350-1000 Hz	1000-1850 Hz	>1850 Hz
Effective order	0	1	2	3	4

Table 1: Effective order of the HOA signals recorded with an EigenMike (approximate values).

The values presented in Table 1 indicate that below about 1850 Hz, the HOA signals corresponding to the order-4 spherical harmonics have very little amplitudes. In order for the energy map to be estimated accurately, the effective HOA order must be taken into account. Therefore in future work the energy map estimation will be implemented in frequency bands, the lower and upper bound of which will be set as a function of the SMA geometry. In each frequency band, only the HOA signals recorded with sufficient amplitude will be used for the energy map calculation.

4.2 Spatial edition of HOA sound scenes

b<>com designed a tool providing basic HOA sound scene editing functionalities, such as adjusting the gain for a specific direction of space or changing the direction of a sound source. This tool was implemented as a VST plugin which we refer to as the *Sound Field Editor*. We describe how the plugin operates below.


Target position
Visualisation of sound field energy
Control position
Sphere
Points
−
+
4 src
Gain setting: residual sig.
Gain setting: control pos.

ID	MUTE	SOLO	JOIN	gain
res	M	S		-5.7
#1	M	S	J	4.1
#2	M	S	J	-3.1
#3	M	S	J	-6.2
#4	M	S	J	5.6

Figure 11: User interface for b<>com's Sound Field Editor VST plugin.

The user interface of the *Sound Field Editor* is shown in Figure 11. In the top part of the interface, a visualisation of the sound field energy is shown, similar to that displayed in the *HoaScope*. On top of the energy map are displayed a number of “control” positions and “target” positions. The plugin functions as follows:

1. For each control position, the plugin estimates a source signal.
2. The contributions from the different sources are subtracted from the scene. The remaining signals are referred to as the residual signals.
3. A gain is applied to the different source and residual signals in accordance to the gain settings displayed in the lower-right part of the interface.
4. The sources are displaced to the target positions (by calculating their new contributions) and mixed with the residual signals.

One can think of different use cases for this plugin. A very simple use would be to modify the gain for a particular direction or region of space, which could be useful in various situations. As an example, consider the case of recording a musical ensemble in a concert hall: in this scenario, it is useful to be able to set the balance between the intensity of sounds incoming from the stage and that incoming from the back of the room. This effect can be achieved with the *Sound Field Editor* by defining one or several control points in the region of interest and setting the corresponding gain(s) until the desired effect is reached.

Another possible scenario where the *Sound Field Editor* is useful is when one desires to modify the perspective in an HOA scene. Recalling the example of the concert hall, a relatively common setup for HOA recordings is to position a microphone array above the head of the conductor, which provides an interesting perspective of the stage. This perspective, however, might be considered as too unusual to be used in the final mix as it is. In this situation, the scene can be modified by defining control positions in the directions of the sound sources and setting target positions in directions corresponding to the desired listening position.

In the current state the *Sound Field Editor* uses a relatively simple method for estimating the signals corresponding to the control positions. This method is suitable in that it requires relatively little computational power. However, it does not separate sound sources very accurately. This can result in audible artefacts in the case of complex sound scenes involving numerous sources. One envisioned improvement of this plugin is to take advantage of the sound field analysis performed in the *HoaScope* in order to improve the quality of the separation. Note that this work will be presented in greater detail in [4].

5 Capture of ambience and reverberation

5.1 Ambient sound capture using a spherical microphone array

To capture ambient sounds, microphones are commonly placed at a relatively large distance from sound sources (e.g., well above the audience or stage). Alternatively, a directional microphone can be used that is pointed away from the sound sources. In case the sound sources move, the orientation of the microphone needs to be adjusted as well, which can be impractical. To mitigate this problem, we designed a solution that automatically suppresses directional sounds while capturing as much of the ambient sound as possible using a spherical microphone array.

Ambient sounds often arrive from all directions simultaneously, and can therefore be modelled as a diffuse sound field. As a starting point, we adopted a beam-former that was designed to capture only diffuse sounds [5]. This beam-former satisfies one or more constraints to suppress directional sounds and one additional constraint that allows the extraction of the diffuse sound without distortion.

For this approach, we use the sound field captured by a spherical microphone array (SMA) with M number of microphones (such as MH Acoustic's Eigenmike). First, the M microphone signals are transformed to the spherical harmonic domain using the spherical harmonic transform. Secondly, the obtained signals are transformed to the short-time Fourier transform (STFT) domain. Thirdly, up to two simultaneous DOAs are estimated per time-frequency using the direction of arrival (DOA) estimator proposed in [6]. Using this DOA information, the beamformer places spatial nulls to the DOAs of the direct sounds, and extracts the diffuse sound. It is important to note that the DOAs are estimated per time-frequency tile, and that many sounds are sparse in the time-frequency domain (i.e., only one or two direct sounds are dominant per time and frequency tile). Therefore, this approach can be used to reduce the direct sounds of more than two moving sound sources. A high-level block diagram of the ambient sound-capturing algorithm is shown in Figure 12. The algorithm has been implemented in MATLAB[6] and is currently being evaluated.

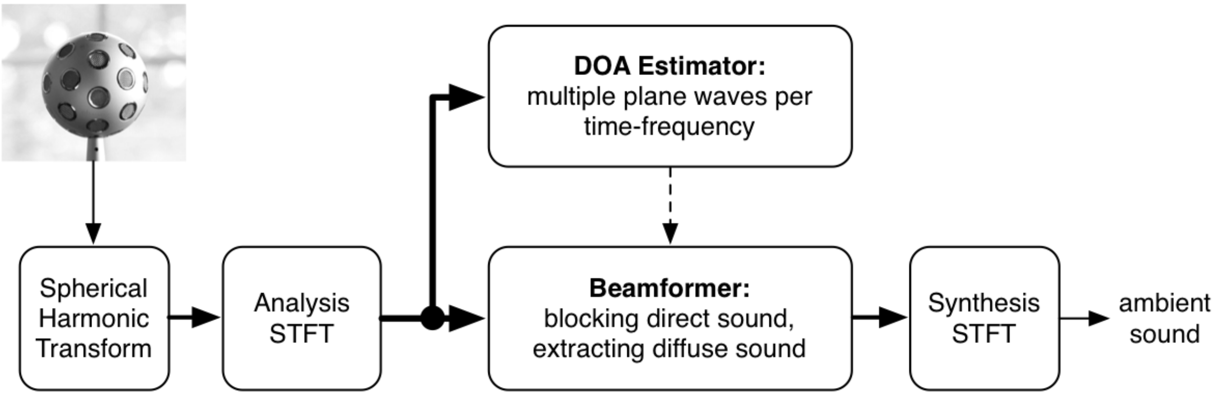

DOA Estimator:
multiple plane waves per time-frequency
Spherical Harmonic Transform
Analysis STFT
Beamformer:
blocking direct sound, extracting diffuse sound
Synthesis STFT
ambient sound

Figure 12: Block diagram of the ambient sound capturing algorithm

[6] MATLAB is a registered trademark of Mathworks

5.2 Analysis and capture of reverberation parameters

Room effects and reverberation are considered as being essential parts of audio post-production and 3D audio rendering. They can be used to simulate a specific room and to achieve a desired spatial sound impression. Representing reverberation in an object-based format is of particular interest for audio broadcasting; it gives the greatest flexibility to adapt the spatial audio scene to the playback device and eases the interaction of the listener with the audio scene. It is, for example, widely used and well accepted in virtual reality and augmented reality applications, where the scene graph modelling software usually provides a geometrical description of the virtual environment, from which sound objects can be easily derived. This information is in general not available in post-production environments for digital broadcast, hence signal-based and parametric approaches are used. The deliverable D3.2 presents in detail methods, workflows, and an example implementation of reverberation processing and storage in object-based broadcast, and it further discusses the advantages and disadvantages of signal-based and parametric approaches.

Signal-based approaches are closely related to usual sound engineering practice. The room reverberation is typically captured with microphone arrays that are positioned in the diffuse field of a room, i.e. at a distance which is larger than the so-called reverberation radius[7]. To avoid that a microphone array partly captures the direct sound, it is common practice to place a spatial null into the direction of the direct sound (see also the block diagram depicted in Figure 12). Alternatively, the reverberation signals can be rendered with reverberation processors during post-production. The channel-based and scene-based approaches then transmit the reverberation signals as separate audio channels, depending on the target output format. This approach is compatible with existing object-based metadata schemes, such as ADM. Figure 13 depicts a general signal-based metadata scheme for reverberation in object-based audio. R0, R1, R2 and R3 refer to the direct sound, early reflections, late early reflections (cluster) and late reverberation, respectively.

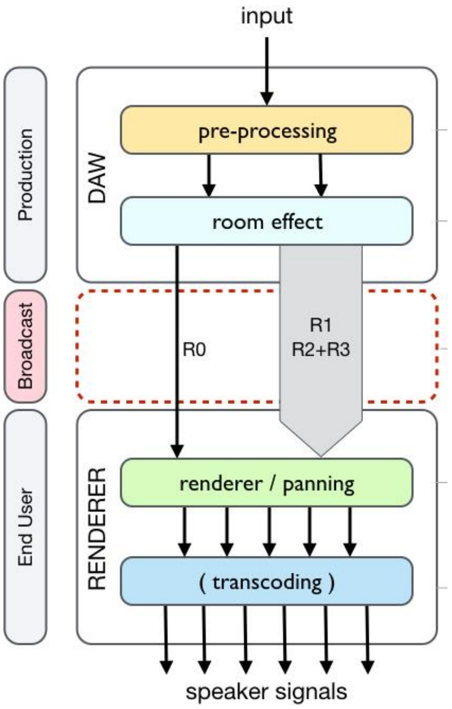


Figure 13: General signal-based metadata scheme for the channel-based and scene-based reverberation on object-based audio.

[7] The reverberation radius or critical distance describes the distance where the direct sound energy diffuse field energy densities are equal [7].

Scene-based approaches decompose the captured reverberation into a set of orthogonal basis functions, typically applying the spherical harmonic transform. The HOA-encoded reverberation signals are transmitted in separate audio channels and then decoded onto a target loudspeaker setup and mixed to the "dry" audio objects. This approach does not require that the target loudspeaker setup is known during the mixing process, as it fully encodes the spatial audio scene. However, the decomposition only gives some limited control over the sound scene in the decoder (e.g., scene rotation, zoom, scaling, and spatial windowing operators are available). In practice, the limited number of microphones and the spatial sampling grid define the maximum applicable spherical harmonic order, and thus the spatial precision and the usable spatial bandwidth for an error-free representation of the sound field [9].

Many end-user playback devices and typical audio transmission channels do only support low-order Ambisonics. Decoding the HOA-encoded reverberation signals onto a lower-order device (e.g., loudspeaker array or binaural audio playback over headphones) requires order truncation to avoid spatial aliasing. Truncating the spherical harmonics order not only results in a reduced spatial resolution (i.e. in an increased localization blur) but also in a high-frequency roll-off and overall energy loss. In [7] we addressed this issue by introducing a spatial blur operator that allows to fade out higher-order components while preserving the overall signal energy. Mathematically, the spatial blur operator is defined as follows:

$$\alpha \rightarrow g_n(\alpha) = 1 - \frac{1}{1 + \mathrm{e}^{-\tau\left(\alpha - 100\frac{N-n+1}{N+1}\right)}}$$

where α denotes the spatial blur factor, τ is a smoothing constant and $0 \leq n \leq N$ the spherical harmonic order. The compensation of the energy loss due to truncation allows for maintaining a constant loudness. This is essential for maintaining a constant signal-to-reverberation ratio during audio playback over different devices, which for instance affects the perceived distance of a sound source.

Figure 14 shows the energy-preserving weighting functions for spherical harmonic truncation up to orders N=5 and the associated order-dependant spherical panning functions. In this figure α denotes the spatial blur factor in percent. For more details the reader is referred to [10].

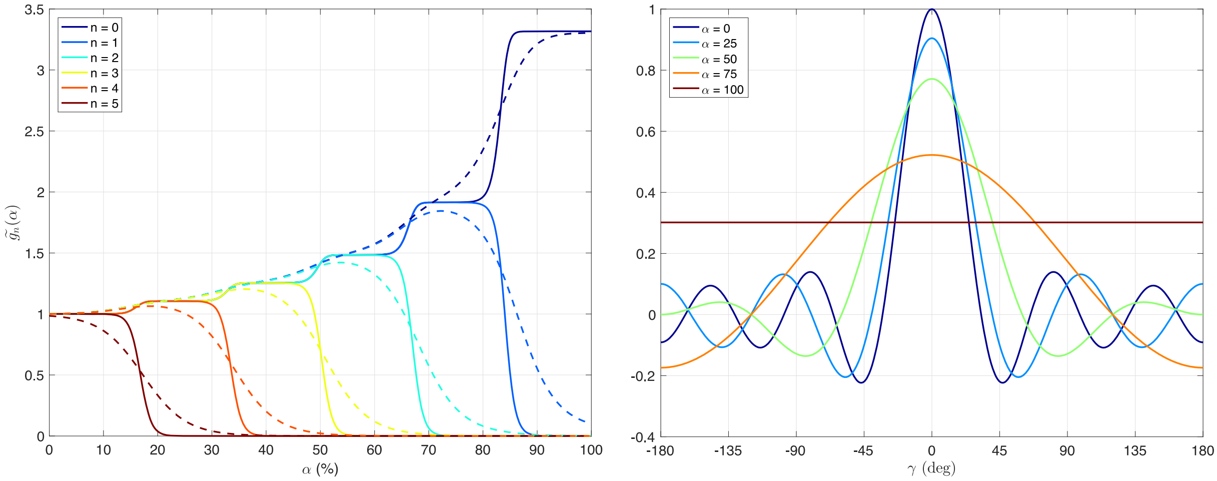
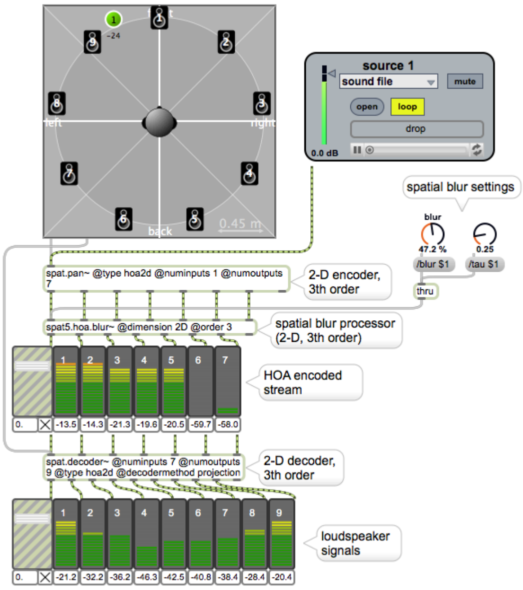


Figure 14: Energy preserving weighting functions (left) and order dependent panning function (right) as a function of the spatial blur factor α (in percent) for spherical harmonic orders up to N=5.

The spatial blur and truncation operator was implemented in IRCAM's real-time spatial audio processing software *Spat~*. Figure 15 depicts a screenshot of the implementation. Without loss of generality, the example shows an HOA audio signal stream of order N=3 decoded to a ring of K=9 surrounding loudspeakers.



spat.pan~ @type hoa2d @numinputs 1 @numoutputs 7

2-D encoder, 3th order

spatial blur settings

/blur $1

/tau $1

spat5.hoa.blur~ @dimension 2D @order 3

spatial blur processor (2-D, 3th order)

HOA encoded stream

-13.5 -14.3 -21.3 -19.6 -20.5 -59.7 -58.0

spat.decoder~ @numinputs 7 @numoutputs 9 @type hoa2d @decodermethod projection

2-D decoder, 3th order

loudspeaker signals

-21.2 -32.2 -36.2 -46.3 -42.5 -40.8 -38.4 -28.4 -20.4

Figure 15: Implementation of the spatial blur function in the IRCAM Spat~ real-time processing environment.

Directional room impulse responses (DRIRs) can be seen as the 3D acoustical fingerprint of a room. They are typically measured with spherical microphone arrays (see e.g. [11]). Convolving an anechoic input signal with a DRIR reproduces the room reverberation and guarantees for an authentic and natural listening experience. However, convolution-based reverberation processing requires a very good signal-to-noise ratio (SNR), which is difficult to obtain under real-world measurement conditions. Within the ORPHEUS project, IRCAM has implemented efficient algorithms for estimating the mixing time[8] and spatial coherence in each frequency band, and for the de-noising of 3D DRIRs. They have been evaluated using DRIRs measured for many source and receiver positions in different venues, including IRCAM's variable concert hall and the Chiesa San Lorenzo in Venice. Figure 16 shows the de-noised DRIR in HOA and spatial domain.

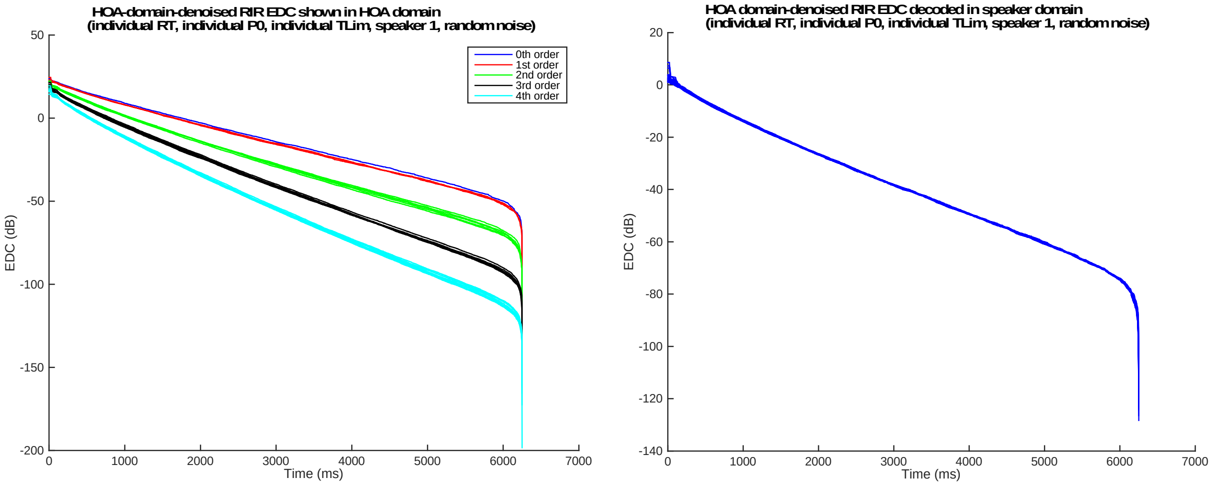
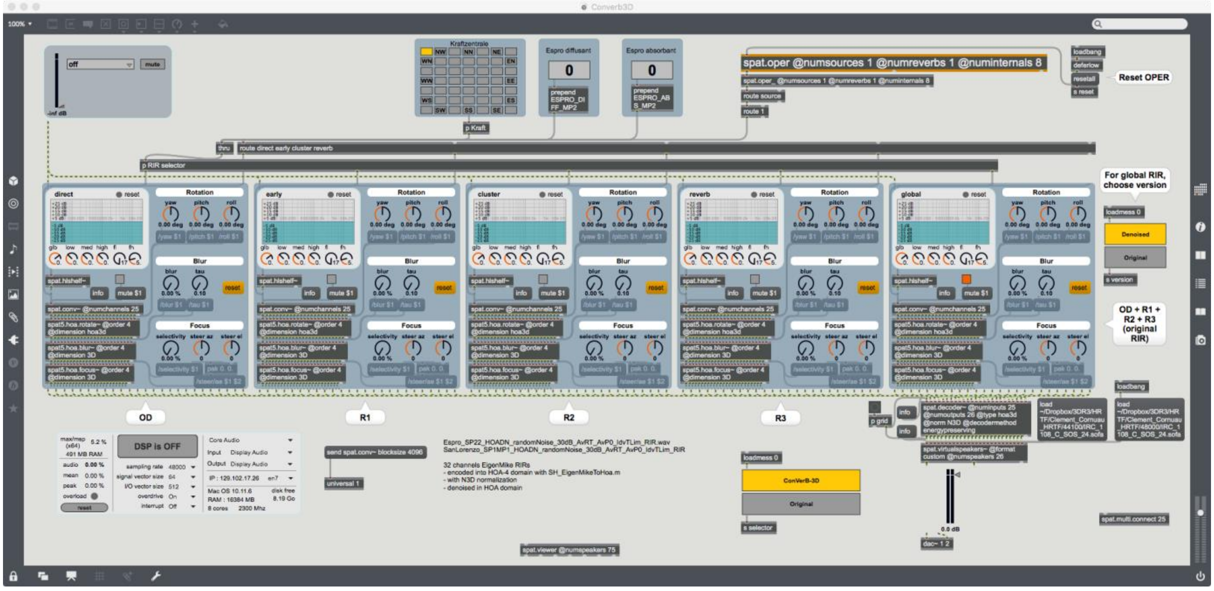


Figure 16: DRIR of the Chiesa San Lorenzo after de-noising in the HOA domain. Energy decay curve of the denoised DRIR in the HOA domain (left) and in the spatial domain (right).

After de-noising, the DRIR can be directly used with convolution-based reverberation processors, without generating any audible artifacts. The notion of perceptual control can be introduced to convolution-based processing by splitting the DRIR into four time segments that correspond to the

[8] The mixing time is the time where the late reverberation (i.e. the diffuse sound field) starts.

direct sound, the early reflections, the late early reflections, and the late reverberation, respectively. The starting point of the last time section is defined by the estimated global mixing time. Figure 17 depicts an example implementation of a DRIR convolution-based reverberation processor in the IRCAM *Spat~* real-time spatial audio processing software.



Converb3D
100%
off
mute
-inf dB
Kraftzentrale
NW
NN
NE
WN
EN
WW
EE
WS
ES
SW
SS
SE
p Kraft
Espro diffusant
0
prepend ESPRO_DIFF_MP2
Espro absorbant
0
prepend ESPRO_ABS_MP2
spat.oper @numsources 1 @numreverbs 1 @numinternals 8
spat.oper_ @numsources 1 @numreverbs 1 @numinternals 8
route source
route 1
loadbang
deferlow
resetall
s reset
Reset OPER
thru
route direct early cluster reverb
p RIR selector
direct
early
cluster
reverb
global
reset
Rotation
yaw
pitch
roll
0.00 deg
/yaw $1
/pitch $1
/roll $1
glb
low
med
high
fl
fh
Blur
blur
tau
0.00 %
0.10
/blur $1
/tau $1
spat.hishelf~
info
mute $1
spat.conv~ @numchannels 25
spat5.hoa.rotate~ @order 4 @dimension hoa3d
spat5.hoa.blur~ @order 4 @dimension 3D
spat5.hoa.focus~ @order 4 @dimension 3D
Focus
selectivity
steer az
steer el
/selectivity $1
pak 0. 0.
/steer/ae $1 $2
For global RIR, choose version
loadmess 0
Denoised
Original
s version
OD + R1 + R2 + R3 (original RIR)
OD
R1
R2
R3
p grid
spat.decoder~ @numinputs 25 @numoutputs 26 @type hoa3d @norm N3D @decodermethod energypreserving
load ~/Dropbox/3DR3/HRTF/Clement_Cornuau_HRTF/44100/IRC_1108_C_SOS_24.sofa
load ~/Dropbox/3DR3/HRTF/Clement_Cornuau_HRTF/48000/IRC_1108_C_SOS_24.sofa
loadbang
spat.virtualspeakers~ @format custom @numspeakers 26
0.0 dB
dac~ 1 2
spat.multi.connect 25
loadmess 0
ConVerB-3D
Original
s selector
send spat.conv~ blocksize 4096
universal 1
spat.viewer @numspeakers 75
DSP is OFF
reset

max/msp (x64) 5.2 %
491 MB RAM
audio 0.00 %
mean 0.00 %
peak 0.00 %
overload

sampling rate 48000
signal vector size 64
I/O vector size 512
overdrive On
interrupt Off

Core Audio
Input Display Audio
Output Display Audio
IP : 129.102.17.26 en7
Mac OS 10.11.6 disk free 8.19 Go
RAM : 16384 MB
8 cores 2300 Mhz

Espro_SP22_HOADN_randomNoise_30dB_AvRT_AvP0_IdvTLim_RIR.wav
SanLorenzo_SP1MP1_HOADN_randomNoise_30dB_AvRT_AvP0_IdvTLim_RIR

32 channels EigenMike RIRs
- encoded into HOA-4 domain with SH_EigenMikeToHoa.m
- with N3D normalization
- denoised in HOA domain

Figure 17: IRCAM Spat~ implementation of a convolution-based DRIR reverberation processor with perceptual control.

Hybrid reverberation processors combine both convolution processing for the early part of the RIR (Room Impulse Respones) and FDN (Feedback Delay Network) for the late reverberation (see [12] for an overview on different algorithms). Applying convolution processing to only the early part (< 200ms) of the DRIR reduces the computational cost and preserves the naturalness and spectral signature of the room response. The late reverberation is typically modelled with computationally efficient FDN, and the model parameters are estimated from the de-noised DRIR. To guarantee for a smooth transition at the mixing time between the two processing stages are crossfaded. The crossfade is depicted in Figure 18.

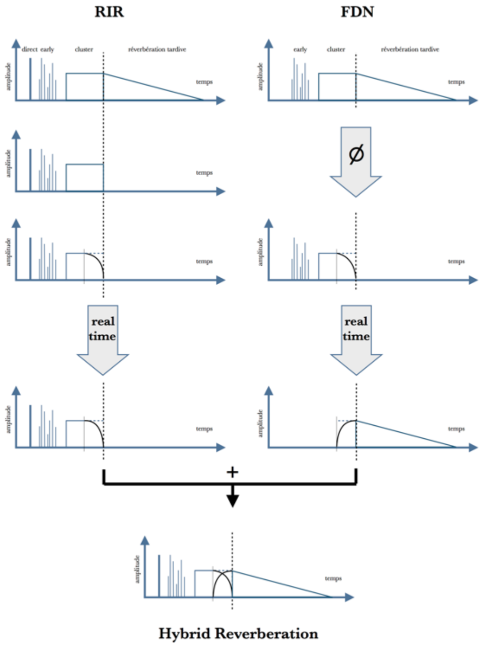


Figure 18: Hybrid reverberation processing with crossfade.

The DOA of the direct sound and early reflections can be estimated from the plane wave density function, which can be obtained from the DRIR [8]. However, the accuracy is limited by the spatial resolution of the spherical microphone array. Improved estimation algorithms are currently under development.

ORPHEUS D3.2, Section 3.1, presents metadata schemes for the representation of reverberation in object-based audio for broadcast, which are in accordance with the ITU Audio Definition Model (ITU-R BS.2076). Convolution-based and hybrid reverberation processors require transmit DRIR data to the end-user application. This is not compliant with the current ADM standard, as it does not allow to send binary data as metadata. However, this can be solved by representing the DRIRs in the AES69-2015 SOFA format. For this reason, we have recently defined a set of conventions named "SingleRoomDRIR", which has been proposed for standardisation. For an ADM compliant transmission of the SOFA-encoded DRIR a new RIFF/BWF/BW64 chunk has to be defined. We propose the following structure:

- Identifier "sofa" (4 bytes)
- Length of data (4 bytes, unsigned little-endian, 32-bit integer)
- Raw data (i.e. the content of the SOFA DRIR file as audio)

The ADM XML description could then directly refer to the name of the impulse response, which is stored in the SOFA file.

An example implementation is currently under development.

6 Conclusions

In this Deliverable a number of tools and methods contributed by the partners during the first half of the ORPHEUS project have been presented. The overarching goal of these contributions was to facilitate the production of audio content in an object-based concept, and especially the production of immersive audio experiences. Two main approaches were taken by the partners. First, we designed methods and tools that enable new possibilities in audio recording, e.g. the possibility to record 3D audio with a 2D microphone array. Secondly, we proposed ways to improve, ease or automate tasks that were previously relatively tedious or inaccessible to non-specialists.

Our hope is that this work anticipates the evolutions of audio content production in the near future. However, it is likely that new practices and workflows will appear which will be driven by content creators. In this regard it is crucial that researchers and technology providers work hand in hand with audio professionals.

References

[1] K. Kowalczyk, O. Thiergart, M. Taseska, G. Del Galdo, V. Pulkki and E.A.P. Habets "Parametric spatial sound processing: A flexible and efficient solution to sound scene acquisition, modification and reproduction," IEEE Signal Processing Magazine, Vol. 32, Issue 2, pp. 31-42, March 2015.

[2] N. Epain and J. Daniel, "Automatic sound source localization for object-based audio recording". Accepted for presentation at the 2017 IBC, Amsterdam, the Netherlands, September 2017.

[3] S. Moreau, J. Daniel, and S. Bertet, "3D sound field recording with higher order Ambisonics – Objective measurements and validation of a 4th order spherical microphone". Proceedings of the AES 120th Convention, Paris, France, May 2006.

[4] M. Hafsati, N. Epain and J. Daniel, "Editing Ambisonic Sound Scenes". Accepted for presentation at the 2017 International Conference on Spatial Audio, Graz, Austria, September 2017.

[5] O. Thiergart and E.A.P. Habets, "Extracting reverberant sound using a linearly constrained minimum variance spatial filter," IEEE Signal Processing Letters, Vol. 21, No. 5, pp. 630-634, 2014.

[6] O. Thiergart and E.A.P. Habets, "Robust direction-of-arrival estimation of two simultaneous plane waves from a B-format signal," Proc. of the IEEE Convention of Electrical and Electronics Engineers in Israel (IEEEI), Israel, Nov. 2012.

[7] H. Kuttruff, "Room Acoustics (4th ed.)". Spon Press, 2000.

[8] B. Rafaely, "Plane-wave decomposition of the sound field on a sphere by spherical convolution," J. Acoust. Soc. Am. 116(4), pp. 2149–2157, 2004.

[9] G. Chardon, W. Kreuzer, and M. Noisternig, "Design of Spatial Microphone Arrays for Sound Field Interpolation," Selected Topics in Signal Processing, IEEE Journal of, 9, pp. 780–790, 2015.

[10] T. Carpentier, "Ambisonic Spatial Blur". Proceedings of the AES 142nd Convention, Berlin, Germany, May 2017.

[11] H. Morgenstern, B. Rafaely, and M. Noisternig, "Design framework for spherical microphone and loudspeaker arrays in a multiple-input multiple-output system," J. Acoust. Soc. Am., 141(3), 2024–2038, 2017.

[12] T. Carpentier, M. Noisternig, and O. Warusfel, "Hybrid Reverberation Processor with Perceptual Control," Proceedings of the International Conference on Digital Audio Effects (DAFx), Erlangen, Germany, pp. 93–100, 2014.

[13] P. Majdak and M. Noisternig, "AES standard for file exchange - Spatial acoustic data file format", Audio Engineering Society Standard No. AES69-2015, New York, NY, US, 2015.

[end of document]