School of Engineering
Electrical and Electronic Engineering

Semester Research Project - LTS2

Responsible Professor: Prof. Dr. Hervé Lissek
Supervisor: Vincent Grimaldi

Design of an Externalized Music Player

Author:
Gloria Dal Santo

SCIPER:
320734

Spring Semester 2021

Contents

1 Introduction

Although listening to music through headphones has become the normality, the sound perceived through these devices appears to come from inside the head. Sounds coming from the real-world are usually perceived as located out the head, or externalized, and their location can in general be resolved. In optimal sound spatialization methods, the delivered sound signal is processed in order to recreate the perception of an externalized sound source whose position can be identified in three dimensions. There are many applications in which sound externalization can be beneficial, among which the improvement of the perceptual realism in hearing-aids technologies, and in augmented and virtual realities.

Convolutional artificial reverberation is a technique that attempts at improving the realism of a sound by simulating the reverberation of a physical space. By convolving a signal with the Binaural Room Impulse Response (BRIR) of a specific environment, one can indeed recreate on the user the same psychoacoustic impact that characterizes a real sound reproduced in that same space.

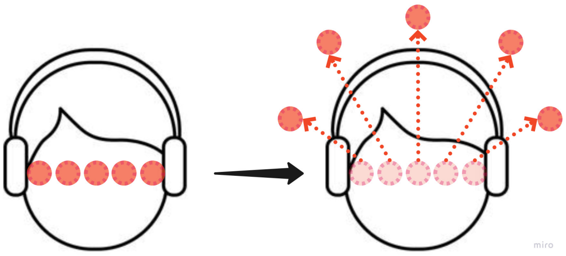


Figure 1: Concept of sound externalization.

1.1 Scope of the project

The scope of this project is that of developing a music player for headphones capable of externalizing the sound according to a specific environment. An intense use of conventional headphones can lead to mental fatigue, resulting in lower reaction time and attention deficits [1]. Externalized sound can help coping with stress and tiredness that stereo and mono sound reproduction systems over headphones induce to the listener. Therefore, it is of great interest to develop a music player that helps at comparing the effects of long-term listening sessions between stereo and externalized sound.

The core of the presented music player is based on convolutional artificial reverberation. For the implementation of the artificial reverberation algorithm, partitioned convolution will be used on a set of BRIRs measured in a listening room at EPFL. The music player will communicate with a motion sensor and use the retrieved information about the movement of the user's head to enhance the realism of the processed sound. It is expected that the listener will be able to localize the sound source in the space as if it was coming from a pair of virtual loudspeakers. The final implementation will include additional tools for the customization of the set of impulse responses in terms of Direct-to-Reverberation energy ratio, length of the impulse response and orientation of the sound source.

2 Theory

2.1 Room Impulse Response

The transfer functions from a sound source to the eardrum are called the **Head Related Transfer Functions** (HRTFs). HRTFs take into account all of the spectral modifications that the original sound undergoes while travelling through the air and when arriving at the ear, head and torso of the listener. They are thus dependent on the direction and distance of a sound source from the listener and on the listener's body itself.

HRTF measurements are usually performed in anechoic chambers and they can be individual or generic. In the first case, the transfer functions are measured with a human as subject with microphones carefully placed at his ear canals. On the other hand, nonindividual HRTFs are measured by means of a dummy head provided with a pair of built-in microphones in both ears.

Binaural Room Impulse Responses (BRIRs) differ from the HRTFs as they are measured in a non-anechoic environment, such as a room or a theater. Moreover, as opposed to HRTFs, BRIRs are described in time domain and contain information about the reflections and reverberations that the sound signal undergoes in its way to listener's ears. BRIRs can be divided into three main parts (Figure 2):

- **Direct Sound:** corresponds to the sound arriving directly to the listener's eardrum, usually related to the shortest path covered by the sound.
- **Early Reflections:** low-density reflections from nearby objects arriving within approximately 80ms (this time interval is shorter for smaller rooms). The arrival time and intensity of the first reflections can have strong effects on the perceived sound. If a reflection arrives within 2ms after the direct sound, it is grouped together with the direct sound and causes an image shift. On the contrary if the time gap between the direct sound and the first reflection is too long, the reflection is perceived as an echo. Strong reflections with short delay can also cause coloration on the perceived sound. Early reflections at around 50ms can support the direct sound by improving speech intelligibility or music clarity [2,3].
- **Late Reverberation:** high-density reflections. Late Reverberations are frequency dependent as they are strongly related to the absorption characteristics of the room and contained objects. In ideal field it is considered diffused and spread uniformly around the room. Late reflections usually smear out the temporal information decreasing intelligibility [3].

Early reflections have a major role in sound localization. In absence of sound reflections, distance is confounded at the ear with intensity, making it harder to assess how far a source is from the listener. Familiar sounds can be localized in an anechoic environment due to intensity cues. However, with unfamiliar sounds, the perceived distance usually converges to a default value [4,5].
In reflective environment there are acoustic cues with one-to-one relationship with distance, such as the **Direct-to-Reverberant energy ratio** (DRR) which can be computed as follows:

$$DRR = 10\log_{10}\frac{\int_0^T |h(\tau)|^2 d\tau}{\int_T^\infty |h(\tau)|^2 d\tau} \tag{1}$$

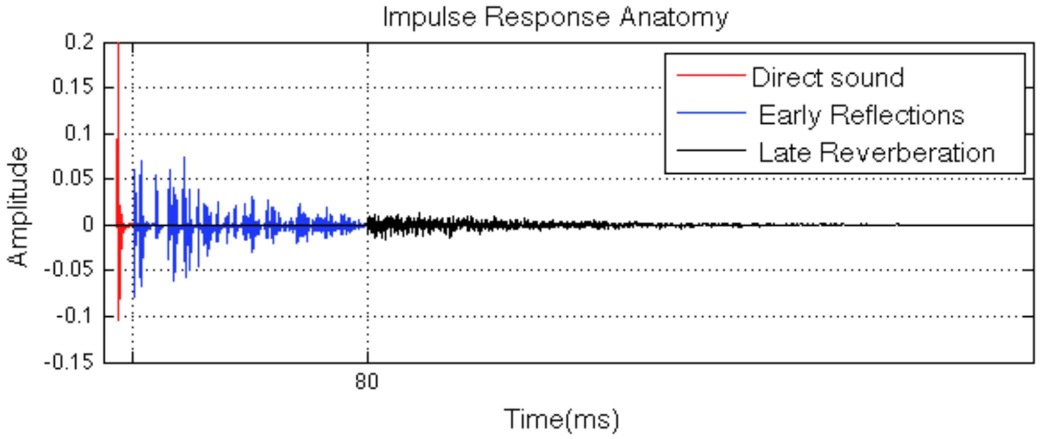


Figure 2: Room Impulse Response (Red: direct sound, Blue: early reflections, Black: late reverberation) [2].

where $h(t)$ indicates the room impulse response and T is chosen such that to separate the direct sound from the reflections (typically $T = 3$ ms).
DRR can be used to estimate the distance of a sound source by exploiting the fact that the direct sound field energy decays with distance while the energy of the reverberations stays approximately constant regardless of the source location (in medium-large rooms).

Direct-to-Reverberant energy ratio have strong influence on different physical variables known to be useful in auditory distance perception:

- **Inter-aural Cross Correlation (IACC):** describes the correlation between signals received at the two ears of a listener. Because reverberation decorrelaties signals at the ears [6], IACC decreases as DRR decreases.
- **Spectral Variance:** refers to the frequency-to-frequency variations in the magnitude spectrum that occur as a result of the interference of reflected sound waves. Spectral Variance increases as DRR decreases [7].
- **Spectral Envelope:** the relative larger absorption of high sound frequency by the air, and by most of the materials commonly used in rooms, modify the spectral content of the reflected sound waves. The spectral envelope of the reverberant sound will thus be shifted towards the lower frequencies as DRR decreases [4].

2.2 Sound Externalization

Sound externalization is the phenomenon that occurs when a sound is perceived as if it originates from a sound source located outside the head. It is considered to be strongly related with distance perception as they share the same continuum, whereby the center of the head represents the minimum distance of an auditory image [8]. As a matter of fact, to subjectively judge sound externalization it is usually asked listeners to grade the presented stimuli over a scale that goes from the center of the head to some fixed distance in the external space. The complementary phenomenon to externalization is internalization (or inside the head localization) and it relates to the idea of violating the listener's expectation about the spatial attribute of the surrounding environment.

2.2.1 Inter-aural Cues

The main cues contributing to the process of out the head localization are the inter-aural cues, that are sound signal features differences between the ears. In sound externalization the most important inter-aural cues are

- **Inter-aural Level Difference (ILD):** it refers to the difference in level between the left and right ears. When the sound arrives from a transversal plane with non-zero azimuth, the shadowed ear will receive a more attenuated sound image compared to the unshadowed one.
- **Inter-aural Time Difference (ITD):** it refers to the difference in arrival time between the left and right ears. The sound has to cover a longer distance in order to reach the shadowed ear, hence it will arrive first to the unshadowed ear. ITD is less important at the higher frequencies, because the wavelength of the sound gets closer to the distance between the ears.

In [9] a headphone synthesis technique has been used to examine the role of inter-aural cues in sound externalization. In the paper it was concluded that ITDs contributed to sound externalization for frequencies below 1.5kHz, while ILDs contributed at all tested frequencies. Externalization information is thus accumulated from spectral levels across the entire spectrum without frequency weighting.

2.2.2 Monaural Spectral Cues (Pinna-related Cues)

Both ITD and ILD are frequency dependent due to filtering by head, pinna and torso, which in turn is dependent on the direction of arrival. Several studies proved this result by either spectrally flattening the HRFTs [10] or by mixing HRTF-filtered stimuli with equivalent stimuli measured using a pair of microphones with the head absent [11]. The direction-dependent filtering by the pinna, head and torso introduces spectral changes to the sound signal entering the ear canal at frequencies above 1-2 kHz [12]. Spectral cues are of great importance in refining the information provided by the ITD and ILD which are limited by the cone of confusion. The **cone of confusion** is a virtual cone that extends outward from each ear and represents the sound source locations producing the same inter-aural differences (Figure 3). In general, if a sound source is on the surface of the cone of confusion, its location cannot be identified using inter-aural cues only [13].

2.2.3 Head Movement

When the listener's head moves relatively to the sound source, the monaural and inter-aural attributes of a signal at the eardrums will change in some particular way. Other attributes will

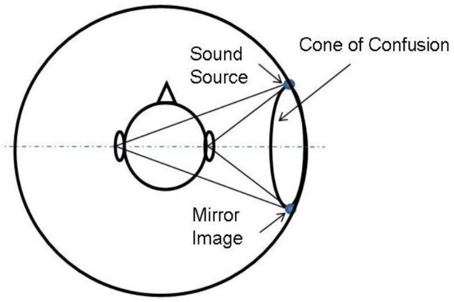


Figure 3: The concept of the cone of confusion [13].

be affected by the head movement, such as sound level and tone color. All these changes facilitate the localization of the sound source and are of major importance, together with monaural cues, when the sound source is on the surface of the cone of confusion [6, 12]. Moreover, in several recent studies [14–16] it has been shown that dynamic cues caused by head movements enhance externalization, especially for frontal and rear sound sources. This can be beneficial for binaural rendering system, as it allows to reduce the BRIR's length by taking advantage of the improved perceived externalization [17].

2.3 Artificial Reverberation

Artificial reverberation algorithms attempt at simulating the characteristics of specific environments. In particular, they aim at reproducing the psychoacoustic impact of various reverberation impulse response features [18]. Artificial reverberation algorithms are employed in many fields, such as music production, hearing-aid technologies and virtual realities. Depending on the application and limitations on the computational complexity, these algorithms fall in one of the following categories:

- **Delay Networks:** the reverberation characteristics are designed into a network of delay lines and digital filters to which the input signal is passed.
- **Convolutional:** the input signal is convolved with a measured or synthesised impulse response.
- **Computational Acoustic:** the geometry of the acoustic space is modelled in order to accurately simulate the acoustic energy propagation in the model itself.

For the presented project, convolutional reverberation is the more suitable. The binaural room impulse responses are taken from a set of measurements performed in a listening room by means of a KEMAR manikin positioned at 2 and 6 meters away from the sound source. The measurements were sampled at frequency of $f_s = 44100$ Hz, at zero elevation angle, and with an azimuth resolution of 10 degrees.

With the availability of a complete set of impulse responses, perceptual sound externalization can be achieved by convolving the dry sound with the impulse response that represent the simulated scenario.

2.3.1 Partitioned Convolution

Partitioned convolution is the most valid alternative to regular convolutions when dealing with long impulse responses and real time implementation. In these situations, direct convolution becomes too computationally expensive and block-FFT convolution leads to unacceptable latency.

In the uniform partitioned convolution algorithm, the impulse response is dividend into P equally-sized blocks with length K samples. Each of these blocks are then zero-padded to the length L, usually a power of 2, and transformed in frequency domain so that to obtain a bank of P filters. The incoming audio stream is processed in partially overlapping blocks of L samples, each starting at $L-K$ samples after the previous. The input data is transformed and multiplied with each filter blocks. The results of the multiplications are stored in P accumulators, each associated with a filter block. The content of the first accumulator is then transformed back in time domain and the latest $L-K$ samples are kept as output stream. Once the following incoming audio block is transformed and multiplied with the P filters, the second accumulator will contain the result of the multiplication of the second filter with the sum of the first and second input audio blocks. The IFFT is then computed on the content of the second accumulator and the latest $L-K$ samples are kept as output stream. The next operations will follow the same mechanism and the output stream will be taken from the following accumulators [19].

When compared to a general overlap-save algorithm, uniform partitioned convolution reduces the latency from $2 \cdot N$, where N is the length of the unpartitioned impulse response, to just L.

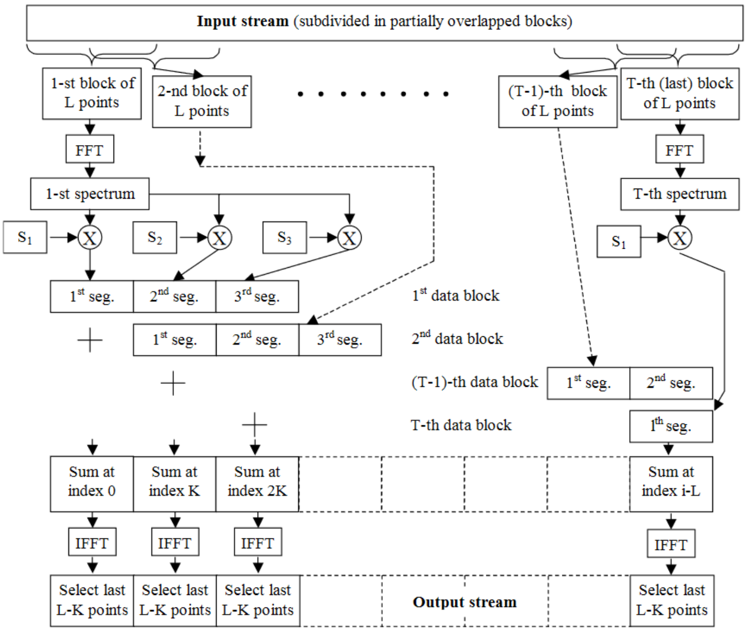


Figure 4: Partitioned convolution [19].

An improved version of this algorithm, here presented for the sake of completeness, consists in slicing the impulse response in non-uniform partitions. The first part of the impulse response is divided into smaller chunks, while the following parts are partitioned in successively increasing blocks. In this way, the most important blocks containing higher energy are convolved and sent to the output stream with lowest latency. The frequency representation of the larger blocks will have a better resolution, improving the perceived externalization when applied in artificial reverberation [20]. In addition, the processing time grows much slower with increased IR lengths than with uniform-sized partitions, thus it is a more scalable algorithm. However, it is much more complex to implement since it requires scheduling and synchronisation of parallel convolution tasks [21].

3 Method

3.1 Project description

The design of a music player for headphones that provides sound externalization has already been the topic of a previous semester project at EPFL. This first implementation was based on MATLAB App Designer. One limiting factor of this design was that all the convolutions had to be calculated at initialization, preventing the music player to change the orientation of the sound source while playing the audio file.

For the presented implementation we have decided to use Pure Data, an open source visual programming environment whose creation was based on the idea of giving to artists and researches an intuitive and user-friendly tool for the development of multimedia applications. The choice on the programming environment was elaborated under the belief that Pure Data is more adequate for real-time audio applications than MATLAB App Designer.

The starting concept underwent to radical changes due to a series of unexpected issues that led to the test and evaluation of different approaches. In what follows, the implementation of a BRIR interpolation algorithm is first discussed. An introduction of Pure Data is then presented in section 3.3, to help the reader to get a clear understanding of the project designs, which are shown in sections 3.4.

Note to the reader: this section includes some technical descriptions on aspects that are related to algorithm design and limitations. The reader that is not interested on these topics can continue from section 3.5.2, where the final design is illustrated.

3.2 BRIRs Interpolation

Binaural Room Impulse Responses, as well as Head Related Transfer Functions, capture the directional dependency of the filtering operation that head and pinna apply on the incoming signal. To maintain the truthfulness of the processed sound, the BRIR with which the dry signal is convolved, has to change at every movement of the head in the three directions. In convolutional artificial reverberation this translates into a potentially infinite number of source-listener positions, each requiring its own measurement. Clearly this is unfeasible, not only measuring the BRIRs at the lowest angle resolution would require a high effort but also it would eventually encounter the limitations set by the device's memory. To overcome this issues, clever interpolation algorithms allows to reduce the required dataset while ensuring perceptually correct spatialization.

While subsequent HRTFs can be interpolated using linear or cubic spline interpolation, in BRIRs the presence of sparse reflections occurring at different times makes these techniques inapplicable. Linear interpolation can result in significant smearing of reflections in the interpolated result, as shown in Figure 5. To bypass this issue, BRIRs interpolation algorithms temporally align the segments containing matching reflections prior to interpolation [22].

Dynamic Time Warping (DTW) is a well known algorithm used for measuring similarity between two temporal sequences and it can be applied to solve the interpolation problem. A warp

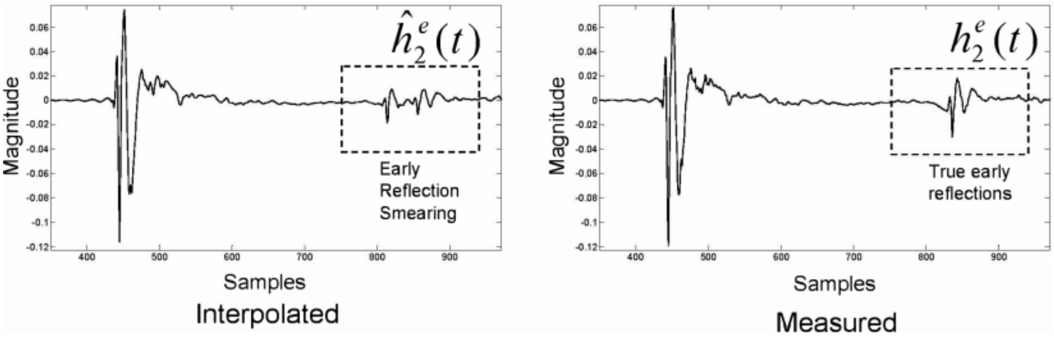


Figure 5: Comparison of impulse response created from linear interpolation between two RIRs to an actual impulse response measured at the position of interpolation [22].

vector is created by identifying the minimum distance path through an accumulated distance matrix. The matrix is constructed by computing the Eucleadian distance between each pair of datapoints in the two impulse responses. As in many DTW-based algorithms, the minimum distance path is subjected to several constraints, among which index monotonicity and continuity. Once the warp vectors are applied on the impulse responses, linear interpolation can be applied. As final step the warped and interpolated vector has to be mapped back into the unwarpped time domain. Although this algorithm has proved to be accurate in medium-large rooms, its use in real-time applications is limited by its high memory demand. As a numerical example, if the DTW algorithm has to be applied on the current project, that would require the computation of a 8912x8912 matrix and several operations on it at every time a movement is detected [22].

An alternative algorithm for BRIR interpolation, presented by V. Garcia-Gomez and J. J. Lopez [23], is based on the assumption that significant information for correct spatialization is contained into few energy blocks. Hence, instead of warping the whole signal as DWT, one could identify these blocks, match them and leave unchanged the rest of the signal. With the project supervisor it has been decided to use this approach for the interpolation algorithm of the music player.
Prior to the development of the algorithm in Pure Data, it has been decided to implement it on MATLAB[1] in order to assess its performance and to adapt it to our needs in a more familiar environment. What follows is a description of the algorithm.

3.2.1 Windowing of the BRIRs

First, the impulse responses that have to be interpolated are split into two part, the initial part h_e containing the direct and early reflection and the second part h_r containing the late reverberation:

$$h_e = h(1:L)$$
$$h_r = h(L - \texttt{ovlp} : \texttt{end})$$

where h denotes the original BRIR and `ovlp` is the number of samples of overlap between h_r and h_e. The overlap is added to avoid discontinuity by applying a crossover between the two segments. In literature, the problem of identifying the boundary point between early reflection and late reverberation does not find a unique solution. A number of criteria have been proposed

[1] MATLAB 2020b, The MathWorks Inc. https://it.mathworks.com/

to find the transition time: some studies used an objective parameter to set the transition 50 or 80 ms after the direct sound, some other studies use the reflection order based on the room dimensions. The interested reader is referred to the work conducted by K. Meesawat and D. Hammershøi for an overview of the main criteria [24].

3.2.2 Dual-band processing of the early reflections

At this point the early reflections of the two BRIRs, h_e^1 and h_e^2, are filtered into two frequency bands. For this purpose, two butterworth 3^{rd} order IIR filters, one low pass and the other high pass, are used. The filters have a cutoff frequency of 150 Hz and they are applied in both forward and reversed direction using the `filtfilt(·)` MATLAB function. With this approach the filtered signals preserve the phase of the original impulse responses, guaranteeing perfect reconstruction when the two signals are added back together. The low band signals are linearly interpolated while the high band signals undergo partial warping before being interpolated. The high band processing is described in the following section.

High Band Processing: The first step is to find the blocks of higher energy that are present in $h_{e,HB}^1$ and $h_{e,HB}^2$ and that are related with each other. To do so, it has been implemented a function that finds the highest peaks, spaced by at least 100 samples, sorts them in descending order according to the value of the peaks, and selects the peaks in $h_{e,HB}^1$ that are most likely to be related with the ones in $h_{e,HB}^2$ in terms of amplitude and time difference by building a distance matrix C as follows:

$$C(i,j) = \frac{1}{(1+\Delta s_{i,j})(1+\Delta p_{i,j})}$$

where $\Delta s_{i,j}$ and $\Delta p_{i,j}$ describe the normalized absolute value of the difference in samples and amplitude. In the original algorithm, prior to computing the matrix, the most relevant peaks are selected according to a weighted average threshold. In the presented project this part is omitted. Instead, based on the known characteristics of the employed impulse responses, only two peaks are selected. At this point blocks of 100 samples centered around the energy peaks are created. A gravity point n_g is then defined as function of the position to interpolate x_{int}, the physical positions x_1 and x_2 on which the BRIRs are measured, and the samples' indexes of the peaks:

$$n_g = \left\lfloor n_1 + (n_2 - n_1) \cdot \frac{x_{int} - x_1}{x_2 - x_1} \right\rfloor \tag{2}$$

where n_1 and n_2 denote the sample index of the peaks in the first and second impulse response respectively (Figure 6). The following steps are based on the assumption that within each block there is a part where the impulse response is concentrated, which must move as a single block, and another part which does not carry relevant information for localization and thus can be stretched. The paper suggested to use a sigmoid function to model the displacement of the samples. However, after noticing that the displacement, defined by $d_{1/2} = |n_g - n_{1/2}|$, rarely takes value greater than 3 samples, it has been decided to use a different approach. The block containing less energy is stretched by adding $d_{1/2}$ in samples whose value is defined by the mean of neighbouring samples. The remaining samples are then shifted by $d_{1/2}$ towards n_g. Finally, the warped blocks are put back into the original signal (Figure 7). The resulting signals can then be linearly interpolated.

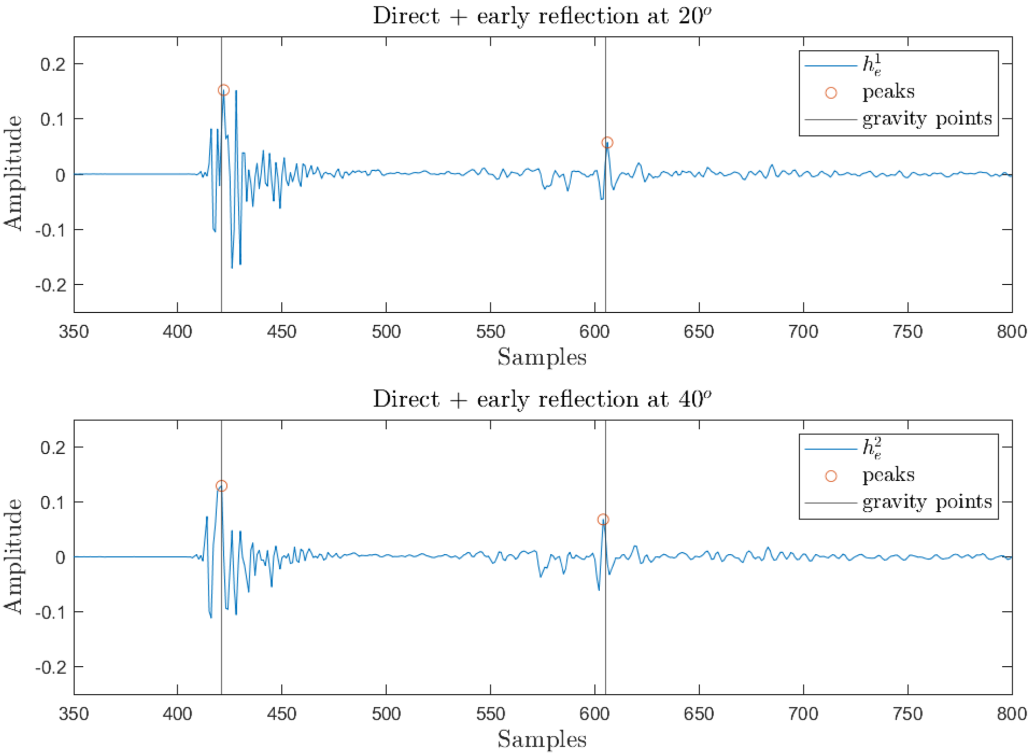


Figure 6: Related peaks and gravity points.

3.2.3 Final mix

The late reverberations h_l^1 and h_l^2 are linearly interpolated and the result $h_{l,int}$ is concatenated to the interpolated early reflection $h_{e,int}$ as follows:

$$h_{int}(i) = w(i) \cdot h_{e,int}(i) + (1 - w(i)) \cdot h_{l,int}(i) \quad \text{for } i = N - \texttt{ovlp}, \ldots, N$$

where N is the length of the BRIR and the weights $w(i)$ are defined as

$$w(i) = \sqrt{\frac{(i-1)}{(\texttt{ovlp}-1)}} \quad \text{for } i = 1, 2, \ldots \texttt{ovlp}$$

3.2.4 Results

The interpolation algorithm has been tested on the interpolation at 30° of impulse responses at 20° and 40°. In Figure 6 it can be seen that related peaks are displaced at most 2 samples apart. Indeed, for the first peak, only the block in h_e^1 has been shifted by one sample toward the gravity point while for the second peak both blocks have been shifted by one sample towards each other (Figure 7). The resulting impulse response is depicted in Figure 8 together with the measured one at 30^o. While the algorithm worked as expected, the final result fails at approximating the measured impulse response. Indeed both the first and second interpolated peaks are shifted

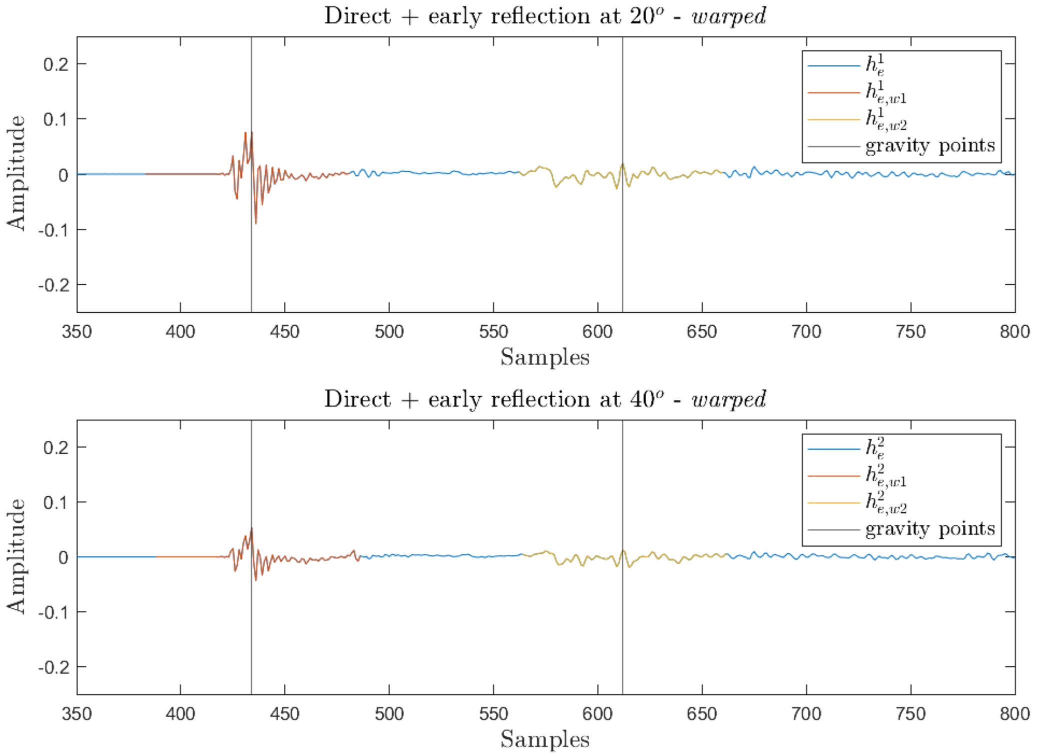


Figure 7: Warped impulse responses.

towards the left, respectively by one and two samples. While the error in the first peak can be justified by the floor function that is applied when computing the gravity point in eq.(2), the displacement of the second peak is somehow unexpected. Because the analysed impulse responses represent the left channel only, the location of peaks are expected to increase with the azimuth angle, as the distance between the speaker and the left ear increases. With the second peak in BRIR(30°) located at a later point in time with respect to the one in BRIR(40°), this monotonic behaviour seems to fail. This inconsistency could be justified by the shape of the room and the orientation of the walls on which the sound signal is reflected. Thus, without any prior knowledge on the room geometry, the interpolation algorithm cannot locate the peaks precisely.

Eventually, further developments of the just presented algorithm have been abandoned as the final design employs direct linear interpolation of HRTFs.

3.3 Pure Data

Pure Data[2], or Pd in short, is an open source visual programming environment developed for creating interactive computer music and multimedia works. Pd gives to artists and researches a tool to create multimedia applications without requiring any knowledge of coding languages. It

[2] Pure Data: https://puredata.info/

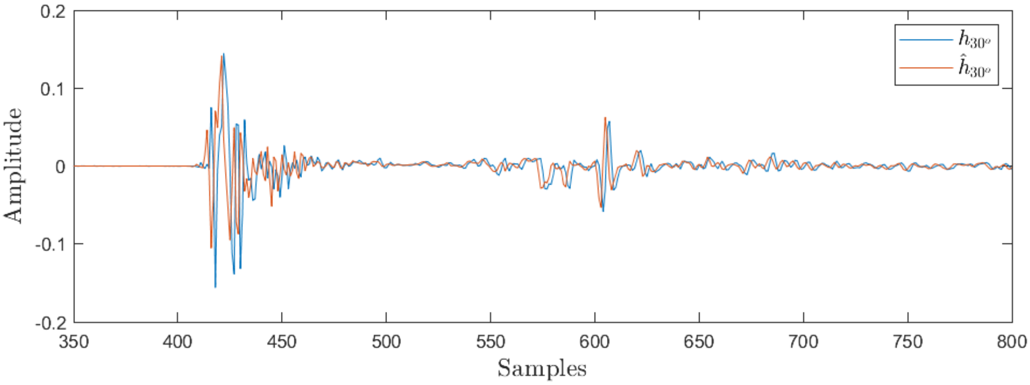


Figure 8: interpolated and measured impulse response.

has been developed by Miller Puckette[3] in the 1990s and is similar to the commercial software Max, also developed by Puckette while he was working at IRCAM (Institut de Recherche et Coordination Acoustique/Musique).

In Pure Data jargon, functions and units follow a specific terminology. It is worth to introduce the description of essential elements in order to ease the reading of the following parts of the report:

- **Patch:** (or abstraction) is the main document in which Pd structures are built.
- **Objects:** building blocks of a Pd patch. Objects were written into the core of Pd program by Miller Puckette and they perform essential algorithmic functions. They receive information via **inlets** and send the result of such functions via **outlets**.
- **External:** is an object compiled from C or C++, not included in Pd-vanilla. They are created and maintained by the Pure Data developers' community.
- **Deken:** is a framework where to create, upload and distribute libraries of externals to make them available in Pd-vanilla.

Audio computations in Pd are carried out by `tilde` objects (e.g. `osc∼`). When audio computation is turned on, Pd sorts all the `tilde` objects into a linear order for running. Then, this linear list is run down in blocks of 64 audio samples each, at 44100 Hz by default (this gives a processing period of 1.45 milliseconds).

Whilst Pure Data vanilla does not include any object that performs fast convolution, two external objects have been made available in the Deken, namely `partconv∼` and `convolve∼`. These implementations of the uniform partitioned convolution have been verified and suggested by the Pd community as a way to create convolution reverb effects when dealing with long impulse responses in real-time applications. In what follows, the structure of `partconv∼` external will be presented by focusing on the main methods. A detailed description of `convolve∼` external will be omitted since its core structure is virtually identical to `partconv∼`'s one. The differences among the two externals is discussed in section 3.3.2.

[3] Miller Puckette's website: http://msp.ucsd.edu/

3.3.1 partconv~

The partconv~ external is included in bsaylor library, developed by Benjamin Saylor[4]. The object requires two arguments: the name of the array containing the impulse response's coefficients and the length of each partition K, which must be a power of 2 of at least 64 samples. The computation of the Fast Fourier Transform (FFT) is based on the FFTW library whose operation is briefly described in the Appendix A.1. The size of the FFT is set to N=2·K requiring each block of incoming audio data to be zero-padded to 2·(N/2+1) samples.

The object updates the impulse response upon reception of a set array_name message which calls the partconv_set method. The tasks carried out by the partconv_set method are:

- retrieval and partition of the array whose name matches the first argument. Each array must be defined in the patch prior to call the method;
- allocation of the buffer of the impulse response and its frequency response. This buffer is unique and must be freed each time a new impulse response is loaded. The memory is allocated using the fftwf_malloc function which guarantees that the returned pointer obeys any special alignment restrictions imposed by any algorithm in FFTW. Once the impulse response's partitions are saved in the buffer and the plan is set, the function fftwf_execute computes the frequency response of the partitions and saves them into the same buffer, overwriting the impulse responses. All the plans have the FFTW_MEASURE flag as argument;
- allocation of the buffer for the DFT of the padded input and definition of the dedicated FFT-plan;
- set up of the circular array of buffers for accumulating the results of the convolution and definition of the dedicated IFFT-plan.

Thus three buffers with corresponding FFT plans are defined. Each buffer is freed at each call of the method if a new impulse response is loaded.

The other method that together with partconv_set builds the core of the external object is partconv_perform, that is the DSP-heart of each signal class. In the partconv_perform a block of input data is allocated in the dedicated buffer and its Fourier transform is computed. Then the multiplication of the FFT of the input block with each BRIR partition is accumulated in the appropriate buffer. Once the buffer has been filled, the inverse Fourier transform is computed. When any one block is inverse Fourier transformed, its main content will be over the first K samples, trailed by low amplitude samples or zeros. A process of overlapping and adding is then performed so that the first K samples of the most recently processed IFFTed block is mixed down with the last K samples of the previous block and is ready to be streamed.

[4] bsaylor library: https://puredata.info/downloads/bsaylor. About Benjamin Saylor: https://puredata.info/author/bensaylor

3.3.2 convolve∼

The convolve∼ external has been developed by William Brent[5] and shares the main features of partconv∼. In addition, it includes a convolve_eq method to balance the frequency components of the output. The main differences between the B. Saylor's external are hidden in the structure of the buffers. To store and accumulate the results of the convolution, convolve∼ does not use a circular array of buffers, instead it uses two separate buffers. Moreover, all the FFT plans have the FFTW_ESTIMATE flag as argument instead of FFTW_MEASURE. This means that the plans might be sub-optimal but the initialization time is shorter as it does not run any computation aimed at building optimal plans.

3.4 Concept

Before designing the project concept, it has been looked for a patch made available by the Pd community to use as a baseline. Consulting the community's official forum, it turned out that there are no previous accessible works that are suitable enough to be used as a baseline. On the contrary, it has been found the earplug∼ external[6], a real time binaural filter that takes a dataset of HRTFs, based on KEMAR impulse measurement, and interpolates the locations in a spherical surface giving to the user control on both azimuth and elevation angles. The earplug∼ external computes convolution in time domain, as the HRTFs are consistently shorter than the BRIRs (128 samples instead of 8192). Despite the fact that it has no direct applications in the starting project design, an analysis of its source code proved to be useful.

3.4.1 Limitations of the available tools

The main idea, discussed with the project supervisor, was that of implementing from scratch a new patch and to use the available externals to perform the partitioned convolution. A first abstraction has been implemented using two partitioned convolution objects (one for each channel). The dataset of BRIR is loaded into separate tables as soon as the patch is opened. In this way, the BRIRs are ready to be called in the set method when the relative message is received. This first implementation underlined the main issue with the objects: as described in the previous paragraph, both convolve∼ and partconv∼ free their buffer upon arrival of the set message (or analyze in convolve∼). This operation may cause the loss of already convoluted audio blocks that are stored into the buffer and are waiting to be sent to the object's outlet. Moreover, the creation of a new plan is computationally expensive and may cause delays in the audio stream: in Pure Data audio processing is scheduled every 64 samples, however messages or external conditions may induce a cascade of messages and operation that will be prioritized and completely run out before the next message or audio block is processed.

To tackle this issues different approaches have been used: the first approach consists in developing an improved version of the partconv∼ external that could be able to load, allocate and transform multiple impulse responses as soon as the patch is opened. It is expected that with this implementation the access to the desired transformed BRIR partitions is simplified,

[5] About William Brent: https://puredata.info/author/wbrent

[6] earlplug∼ download page: https://puredata.info/downloads/earplug

at the expense of a consistent increase in initialization time. The second approach uses multiple `convolve~` running at the same time on subsequent BRIRs and a logic that switches between them when the head movement is detected.

3.5 Design

3.5.1 Initial Concepts

3.5.1.1 Design 1: `partconv~`

It has been decided to work on `partconv~` instead of `convolve~` because of its simplicity and clearer structure. The main objective was that of modifying the `irbuffer` buffer that stores the frequency coefficients of the BRIR in such a way that a new incoming BRIR could have been transformed and stored without causing the loss of the ones already allocated in the buffer. Moreover, in this way the FFTW planner of the impulse responses has to compute the ideal plan only at the beginning, preventing further delays. To achieve this, the `irbuffer` has been modified into a circular buffer which is filled up with the incoming impulse responses. Impulse responses are loaded through the same `set` message used in its native version. To load the entire dataset a sequence of `set` messages has to be sent to the external.
After applying these changes on the source code, the updated external has been tested on a simple testing patch. Even though the C compiler did not highlight any relevant error while compiling, once the first `set` message was received, Pd crashed and became unresponsive. Small adjustments have been applied but none of them seemed to improve the external. Without an intuitive way of debugging the external in Pd, it was difficult to detect the error causing this behaviour. The cause of this crash could be in the initialization phase of the buffer, whose memory has to be allocated and on which one FFTW plan for each BRIRs has to be defined, causing a computational overload.

As an alternative, a more brutal approach has been taken. This consisted in ignoring the routines that free the buffers and destroy the FFT plans. To not deallocate the memory may lead to memory leaks which can accumulate and in turn lead to a used-up of the resources. Despite this drawback, the output of the external improved noticeably when compared to the native version. However, the output audio stream still presented clicks in conjunction with the reception of a `set` message. Similar artifact arose also in the second design.

Several mods based on the just discussed implementation and some new ones consisting in loading the impulse reposes through a header file have been tested. Because none of these mods improved the result, and in agreement with the supervisor of the project, it has been decided to attempt a different approach that does not require C programming. The development of complex Pd externals has proven to require coding experience and sufficient confidence with the Pd debugging environment, for which a skillset outside of the field of digital signal processing is needed in order to obtain significant results.

3.5.1.2 Design 2: multiple `convolve~`

To avoid the issues encountered during the development of the external, in the current design only Pd-vanilla objects and verified externals have been used. The patch consists of a circular chain of 5 `convolve~` externals per channel running at the same time. The outlet of each external is connected to a multiplexer object that is controlled according to the output of the head tracking system. With this configuration it is always ensured that the externals running the desired BRIRs and the two closest ones are always ready to be used. As soon as the patch learns from the head tracker that the azimuth angle has been decreased or increased by one step, the `convolve~` external that is farther in the chain will be updated with a new BRIR, consistent with the direction of the movement and its position in the chain. Then, the logic that controls the multiplexer selects the next `convolve~`'s outlet, which will correspond to the new azimuth angle. In this way the output of the multiplexer will never coincide with the newly updated `convolve~` avoiding any artifact caused by the reception of the `analyze` message. In the initial phase, the head tracker is simulated with a sequence of messages with period 0.5 s that goes periodically from −90° to 90° and back to −90° with an increasing step of 10°.

A first audio test of the patch exhibited audible artifacts, similar to those experienced with the first design, and which can be describes as ”pops” or ”clicks”. These artifacts are neither followed nor anticipated by any dead time, on the contrary the ”clicks” seem to overlap the audio stream. Figure 9 depicts the spectrum of the audio stream that comes out from each convolution, together with the output of the multiplexer (bottom). The intensity has been boosted by a scaling factor of 20 dB/dec in order to highlight the spectral lines in the high frequencies. It can be observed that the artifacts in the bottom track coincides with the update of a `convolve~` object, even though the output of the multiplexer corresponds to the output of a `convolve~` object that is not being updated. Although, the intensity of the unwanted spectral lines is not uniform, after a listening session of a longer audio stream it became clear that the ”clicks” are more evident when the multiplexer's selector jumps from the last to the first inlet. Figure 10 depicts the spectrum of the audio stream that comes out from each convolution, together with the output of the multiplexer (top) when this is set to stream the signal in the first inlet (*convolve1* in the figure). The configuration is similar to the previous one apart the multiplexer's selector being inactive. In this case it can be seen that the spectral lines are less uniform and that no additional spectral lines appear when updating the other `convolve~` objects.

The following approaches have been adopted to understand and to prevent the occurrence of the artifacts:

- introduce larger delay from the moment that a `convolve~` object is updated to the moment at which the multiplexer's selector set the next inlet. This adjustment may help preventing conflicts between the different tasks that the dsp has to carry out within a processing period;
- avoid the passage from the last to the first inlet in the multiplexer;
- reduce the number of convolutions running simultaneously to reduce the computational complexity;
- reduce the length of the BRIRs.

None of these seemed to improve considerably the resulting audio stream. Although the artifacts have been attenuated, their presence was still audible.

The problems encountered throughout the advancement of these implementations have been discussed with two members of the Pd developers' community[7] [8]. The conclusion that can be drawn from our discussion is that the intense use of the FFT planner from `convolve~` may cause spikes in the CPU usage and the rescheduling of the operations listed in the dsp-tree. In conclusion, to provide partitioned convolution with the possibility to change the impulse response in real-time would require a complete revision of the source code.

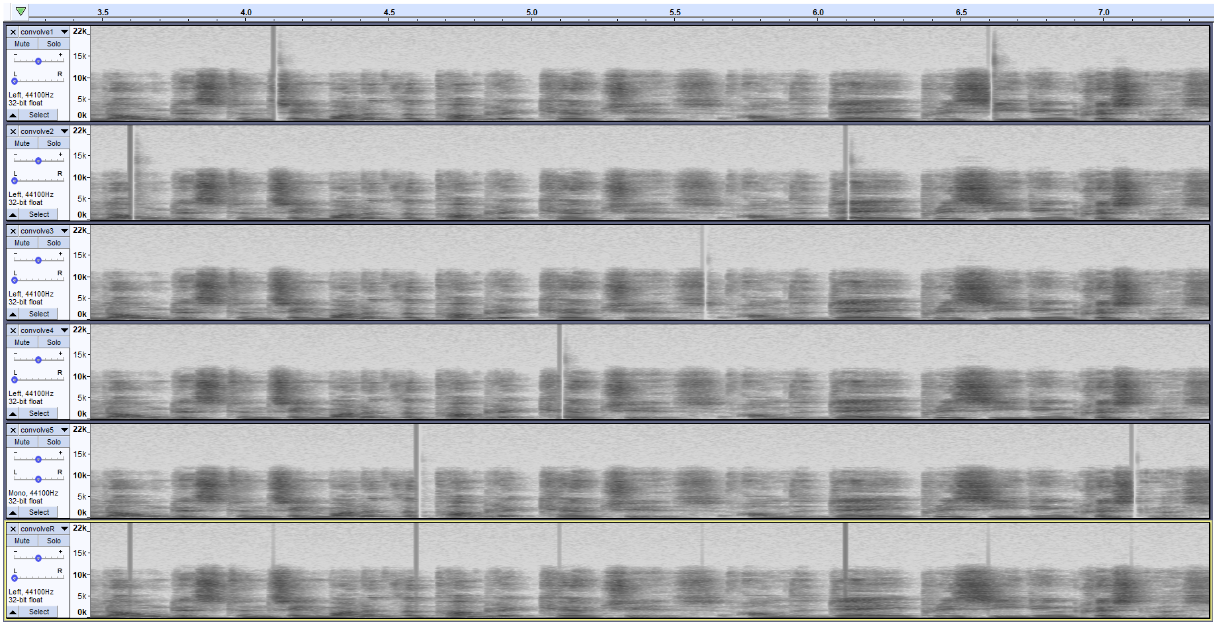


Figure 9: Spectrum of the right channel (Design 2).

3.5.2 Final Design

3.5.2.1 Design 3: `earplug~`

The last and final design is based on the `earplug~` external written by Pei Xiang[9]. `earplug~` is a real-time binaural filter, based on KEMAR impulse responses, that convolves the incoming audio stream with the required HRTFs according to the specified azimuth and elevation angles. In `earplug~`, complete coverage of the azimuth angle and a control of the elevation angle from -40^o to 90^o is provided by linearly interpolation of the HRTFs measured in 366 positions distributed on a spherical surface[10]. The main advantage of this external is that it does not

[7] Email correspondence with: Lucas Cordiviola, on April 8th. Lucarda: https://github.com/Lucarda, and https://puredata.info/author/lucarda

[8] Email correspondence with: Katja Vetter (ktjia), on May 3rd. katjia: http://www.katjaas.nl, and https://puredata.info/author/katjav

[9] Pei Xiang: pxiang@ucsd.edu

[10] HRTF measurements of a KEMAR dummy head microphone: compact dataset https://sound.media.mit.edu/resources/KEMAR.html

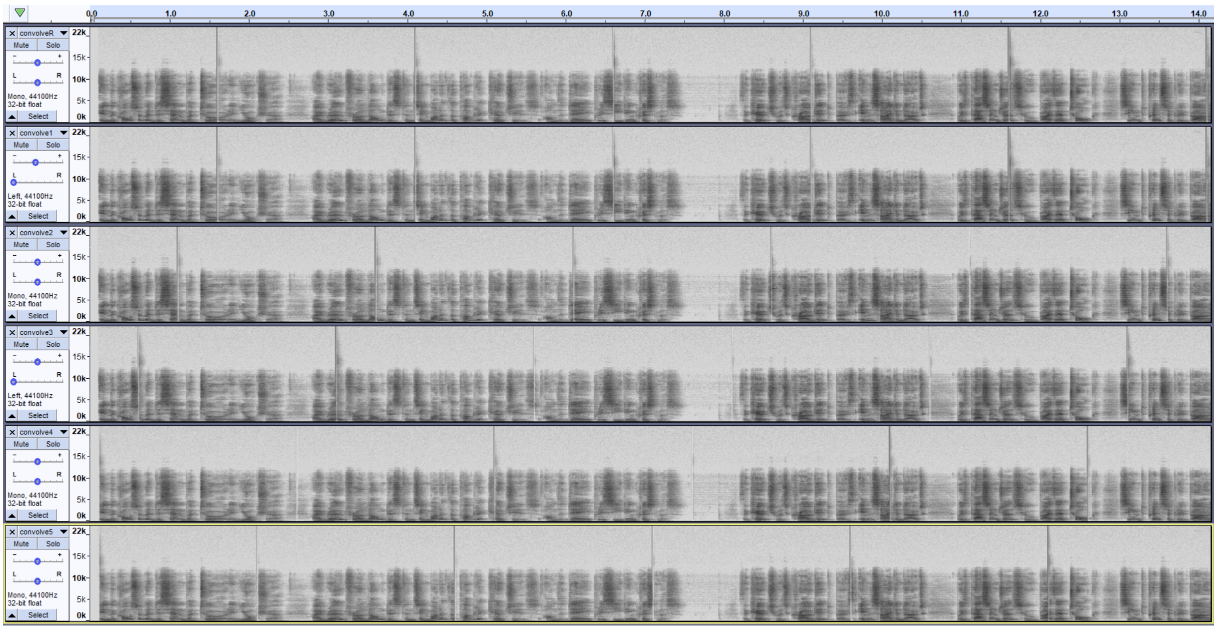


Figure 10: Spectrum right channel, static multiplexer (Design 2).

require Fourier transformation nor interpolation algorithms based on time warping, reducing considerably the computational load.

Although they provide good sound localization, HRTFs fail in externalizing the sound, making the `earplug~` external not suitable for the main scope of the project. To give a more accurate perception of the distance between the virtual sound source and the listener, sound reflections have to be added to the object's output stream. Following this idea, it has been designed a patch based on two `earplug~` objects, one for each channel, and four `convolve~` objects for sound externalization. For the result to be perceptually convincing and to avoid artifacts, the room impulse responses used by `convolve~` has to contain only the early reflections and late reverberation. The limitations of this design are once again set by the partitioned convolution: the ability of `earplug~` to cover the complete azimuthal plane can not be fully exploited, as it would require the update of `convolve~` externals with a new impulse response, encountering again the artifacts experienced in designs 1 and 2. Under this constraint, it has been decided to fix the early and late reflection to a constant location and to limit the coverage range of the HRTFs. If, however, the discrepancy between the HRTFs and the reflections is too large, the sound externalization is expected to fail. In this regard, a previous study on BRIRs' grid resolution has suggested a maximum grid of 5° × 5° for horizontal and vertical head movements [25].

A block diagram representing the main components of the music player in the default configuration is depicted in Figure 11. The HRTFs are set to ±30° azimuth and 0° elevation, to simulate a typical two loudspeakers setup positioned in a equilateral triangle. The early reflection and late revrberation (ER+LR) are also set to ±30° accordingly. Each channel of the the incoming dry signal is convolved four times in total, two in time domain by the `earplug~` and two in frequency domain by `convolve~`, which applies partitioned convolution. To avoid a doubling of the direct sound, the ER+LR are obtained from the available BRIR dataset by aligning the initial delay with that of the HRTFs and setting to zero all the samples related to the direct sound (Figure 12). The head tracking system controls the azimuth angle of the direct sound. Since the ER+LR are not being updated, the output of the head tracker has been limited to ±45° with respect to the median plane. The outles of the `convolve~` objects are multiplied by a scaling factor $\alpha_{L/R}$ which controls the Direct-to-Reverberation energy ratio. The coefficients $\alpha_{L/R}$ are computed according to eq.(3) where h_D and h_R correspond to the HRTF and ER+LR respectively, both at 0° azimuth angle, and DRR is set by the user.

$$\alpha_{L/R} = 10^{\frac{-DRR}{20}} \cdot \sqrt{\frac{\sum_{n=1}^{N} |h_D^{L/R}(n)|^2}{\sum_{n=1}^{N} |h_R^{L/R}(n)|^2}} \tag{3}$$

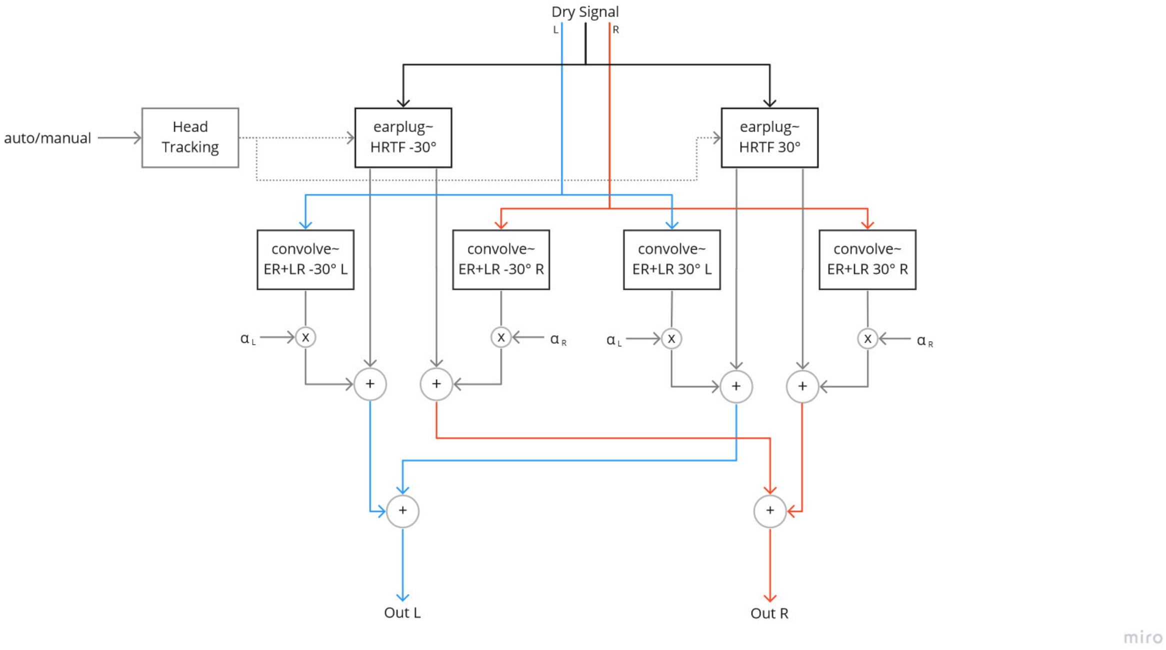


Figure 11: Diagram of the sound externalization block.

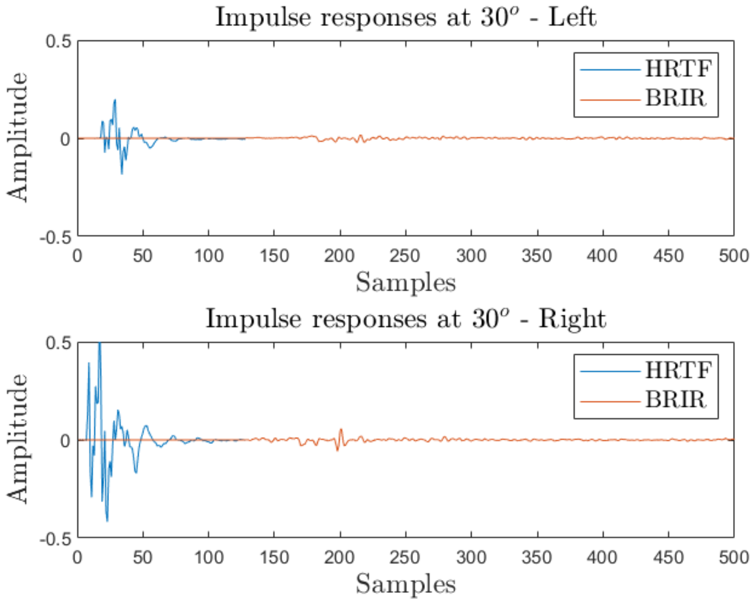


Figure 12: HRTFs and modified BRIRs at 30°.

4 User Interface and Functionalities

The functionalities that can be accessed through the Graphical User Interface (GUI) of the final project design, shown in Figure 13, can be divided into three components: music player (top left), head tracking system (top right), and artificial reverb for sound externalization (bottom).

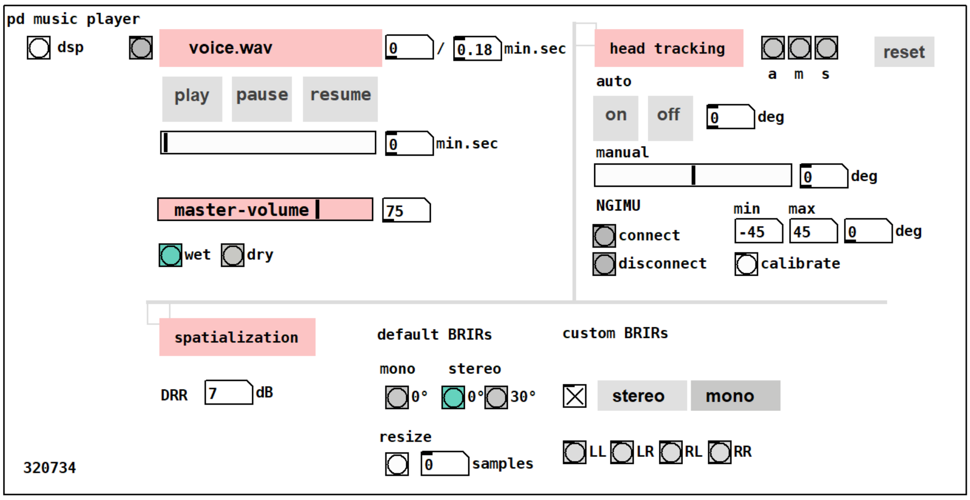


Figure 13: User interface of the music player.

4.1 Music Player

In this section the user can load the sound file from the file explorer by clicking on the button positioned on the left of the filename. The `play` button plays the selected track from the beginning. The `pause` button pauses the audio stream at any time, while with the `resume` button the player can be re-activated from the same time stamp at which it was stopped. The horizontal slider immediately below the previously mentioned buttons enables the user to scroll through the audio file.

The final elements of the music player are the master volume horizontal slider labeled as such, and a pair of buttons labelled `wet` and `dry` that enable the user to switch between the dry and the processed signals directly. This functionality is of particular interest for comparing the perceptual externalization provided by the processed signal with the non externalized sound.

4.2 Head Tracking System

In this section of the GUI, angles are displayed in number boxes. They refer to the position in the azimuthal plane of the listener's head with respect to the loudspeakers setup (Figure 14a). The user then has the possibility of enabling the `auto` mode, as opposed to a manual mode. In `auto` mode, the music player simulates the back and forth rotation of the head from 45° to $-45°$ at a default period of 2 seconds. In `manual` mode, the user can set the desired angle using the horizontal slider labelled as such.

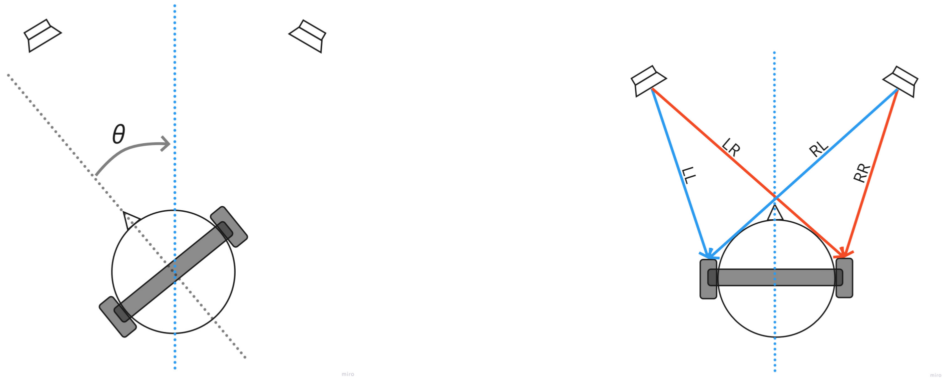
The head tracking system is integrated with a subpatch that enables communication with a IMU tracking[11] device, and retrieves from it the information about the yaw Euler angle of the user's head. Data is transferred over WiFi between the device and the computer on which Pure Data is running. Using the connect button, the user can enable the communication with the sensor. The device can then be positioned at the rest location and calibrated with the dedicated button. One can set the minimum and maximum angles that can be detected and over which the sensor's output remains fixed.

To switch from one mode to the other, one can use the three buttons positioned on the left side of the `reset` button (`a` - auto, `m` - manual, `s` - sensor). The `reset` button deactivates the simulator, disconnects the patch from the NGIMU, and sets the slider to 0.

4.3 Artificial Reverb

This last section of the GUI controls the objects relative to the sound externalization system. Using the `DRR` number box, the user can set the desired Direct-to-Reverberation energy ratio (set to 7 dB by default). The scaling factors $\alpha_{L/R}$ are then computed according to eq.(3). The patch is capable of simulating both the mono and the stereo loudspeaker set up. In the mono set up, at rest both HRTFs and ER+LRs refer to the 0° angle. In the stereo setup the HRTFs always refer to two virtual loudspeakers located at ±30°, while through the dedicated buttons one can select whether to use the ER+LRs at 0° or at ±30°.

The patch allows the users to load their own set of ER+LRs, by first selecting whether a stereo or mono loudspeaker set up is desired. To load the impulse responses the user can use the buttons below the `stereo/mono` selector. The audio convolved with the impulse responses loaded in `LL` and `RL` will be added to the `earplug~`'s left channel, while the audio convolved with the impulse responses loaded in `LR` and `RR` will be added to the right channel (cf. to Figure 14b). Only `LL` and `RR` are required if mono is selected.



(a) Head tracking system angle convention.

(b) BRIRs naming convention.

Figure 14: Conventions used in the spatialization system.

[11] NGIMU, x-io Technologies: https://x-io.co.uk/ngimu/

5 Listening Test and Discussion

Externalized sound could potentially be of great importance in reducing the mental fatigue that stereo sound reproduction leads to after a prolonged listening session. At this point in the project, a subjective psychoacoustic test can help in assessing the performance of the system on both perceived externalization and psychic effects. However, due to the COVID-19 pandemic it was difficult to setup reliable tests on site, which would have taken place in the same listening room where the BRIRs were previously measured. Even to conduct an informal remote psychoacustic test was impossible, as it would have required to recruit several listeners willing to install Pure Data on their laptop together with all the required external libraries, a process that may be complicated for new users.

With these considerations, it was preferred to document the personal feedback of the author of this paper, limited at assessing the perceptual effects of different spatialization parameters. In all these listening sessions the subject knew which parameters were changed as she was in complete control of the music player. The sessions were conducted using a dedicated sound card and studio-quality headphones, in order to avoid the introduction of artifacts related to low quality systems that may bias the result of the test. Knowing that the subject was in a room smaller and acoustically different from the one where the BRIRs were measured, it was expected that the perceived externalization would have been more incline to fail.

The baseline from the tested configuration was the following:

- ER+LR were taken from the data-set measured at 2 meters from the loudspeaker;
- Direct-to-Reverberant energy ratio set to 7 dB;
- the default set of BRIRs measured at ±30° was used on the stereo set up;
- the BRIRs were full-length (185.8 ms);
- the head tracking system was in the OFF state. None of the available head tracking options were active.

Under the baseline settings, the reproduced sound was perceived externalized, although slightly closer than the actual location of the virtual loudspeakers. The sensation was that of being in a closed small room, as expected. The process of switching between the wet and dry sound, highlighted a timbral change in the middle-low frequencies of the played track, while higher frequencies remained intact.

5.1 Effect of the Direct-to-Reverberant energy ratio

The DRR controls the amount of reverberation that is present in the processed sound. In this test it was expected to perceive a decrease in externalization as the DRR increased. With DRR set to 0 dB, the perception was indeed that of a more diffuse and open sound, somehow overwhelming on the frontal part of the head. Increasing the value of the DRR, the sound seemed to converge to the medial frontal plane. At 10 dB, the sound was still perceived outside the head, although it was closer to the face especially for high frequencies. Finally at 18 dB the sound was perceived inside the head, as in the dry sound. In terms of frequency content, the audio was lacking in the low frequencies.

5.2 Effect of different BRIR combinations

In this test the three available BRIR combinations were tested. These combinations are: stereo set up with ER+LR at ±30°, stereo set up with ER+LR at ±0°, and mono set up centered at 0°. The results of this test could help evaluate whether a configuration requiring less memory (like the second one) can be used to approximate a more accurate but memory expensive configuration. With the first configuration the sound was perceived as being more diffused and open than the other two. With the other stereo configuration, the sound source was perceived as being shifted towards the center, but still clearly externalized. The reverberation was also perceived as being less diffuse than with the other set up. Finally, with the mono set up, the sound was still perceived as being externalized, although the virtual sound source seemed to be more distant. Moreover, the reverb was perceived as coming from a cone, centered in the median plane and extended towards the head.

5.3 Effect of BRIR length

Starting from the length of the original BRIRs (186 ms), the number of samples has been divided by 2 at each listening test. With 93 ms the perception of externalization remained unchanged, while for 47 ms the sound was perceived slightly dryer and less externalized. For 12 ms samples the source was still outside the head but it was perceived closer to it. Halving once again the number of samples, only the first early reflection was kept, resulting in sound perceived almost inside the head. For a length below 6 ms the sound source was always perceived inside the head.

5.4 Effect of the head tracking system

The head tracking system performed well as it did not introduce any perceivable latency. Throughout the test, the BRIRs were fixed, while the HRTFs were changed according to the azimuth angle of the listener's head (limited between $-60°$ and $60°$). Despite the fact that, in the range $[-45°, 45°]$ the virtual sound source was perceived relatively fixed and still, when the listener's head moved outside this range the perception was that of a moving source, making it harder to define its location. The sound was always perceived frontal (to a small extent at least) when the head was at the extremes of the the limiting range.

5.5 Discussion

It is worth highlighting that the results reported in the previous sections were drawn from the feedback of the author on listening sessions performed in a room different from the one described by the BRIRs. Moreover, no test on mental fatigue was performed because of a lack of adequate equipment and time. Formal tests on mental fatigue are usually assessed using Visual Analogue Scale (VAS) [26], magnetoencephalography (MEG) [27], and eye-tracking measurements including indices associated with pupil measures, blinking, and oculomotor-based metrics [28].

Although the setting of the test was not optimal, the results seem to be in conformity with what has been found in previous studies. On the length the impulse response, many studies have shown that the reverberation after about 80-100 ms does not influence externalization judgment, [29, 30], explaining why the listener did not perceive any change when moving from 8224 to 4122 samples, 186ms and 93ms respectively.

The results from the test on variable DRR also seem to be in agreement with previous study. In research by Zahorik, et al. [31] it is suggested that, because of the inverse-square law on the direct sound, closer sources produce a greater portion of direct-path energy relative to the amount of reverberant energy than do farther away sources. Moreover in research by Mershon, et al. [32] they demonstrated that listeners' judgments of apparent source distance are much more accurate in a reverberant environment, thus at lower DRR, than in an anechoic environment. For what concerns the head tracking system, according to what was discussed in **??** an enhanced externalization for frontal sound sources was indeed expected. However, because of the limitation imposed by a fixed ER+LR's orientation, moving the head more than $10-15°$ resulted in wrong localization. This results in disagreement with the study in [25], which suggested a BRIR grid resolution of at most 5° to ensure sufficient externalization. On the contrary, in the presented project only the ER+LR where fixed, while the direct sound was linearly interpolated in a continuous way, which may explain why the perceived externalization was satisfactory even at larger head movements [29].

6 Conclusions and Future Work

In this paper the design of an externalized music player was presented. Throughout the development several limitations and difficulties were encountered, leading to a redesign of the final concept. All the tested designs were based on convolutional artificial reverberation and a set of previously measured HRTFs and BRIRs.

The advantage of the presented music player, is the provided control over several parameters that can be used to investigate the effect on perceived externalization of different configurations. These are the Direct-to-Reverberant energy ratio, the BRIR's length, and the possibility to load a different set of early reflections and late reverberations. Moreover, the music player allows the user to connect with a IMU motion sensor for head tracking, improving considerably the realism of the simulated virtual source. A feedback from the author of the paper was also reported, which proved to be in agreement with what had been previously shown in literature.

The presented project represents a good starting point for future implementations. Several aspects of the current design may be improved, in order to fulfill the main scope of the project:

- the music player can be used only through Pure Data environment, imposing strong limitations on its distribution. In Pure Data it is not possible to compile a patch as a binary program so that it can be run as a stand alone application. Translating a Pd patch into Max/MSP[12], which can be thought of as the commercial version of Pure Data, could solve this problem. Max/MSP is capable of running patches as standalone applications and it gives more freedom in the GUI design so that a more user friendly application interface can be created;

- the current design is the result of several approximations of the starting design that were needed in order to deal with the limitations imposed by the available Pd externals. Hence, it could be of great interest to develop a Pure Data external that improves the ones already available by giving the possibility to change the set of BRIRs in real-time. With such an external, the starting design can be implemented, and there will not be a need to separate the processing of the direct sound from the processing of the early reflections and late reverberation;

- finally, one further step would be that of conducting a formal psychoacoustic test to assess the effect of externalized sound on mental fatigue. For the results to be of academic relevance, the measurements should be performed on a larger sample size, in a controlled environment, and should follow the procedure of formal listening tests on perceived quality and psycho acoustic effects.

[12] Max/MSP, Cycling '74: https://cycling74.com/

References

[1] MA Boksem, TF Meijman, and MM Lorist, “Effects of mental fatigue on attention: An ERP study”, *Cognitive Brain Research*, vol. 25, no. 1, pp. 107–116, 2005.

[2] F Georgiou and B Fazenda, “Relative distance perception of sound sources in critical listening environment via binaural reproduction”, *School of Computing, Science and Engineering University of Salford, Salford, UK*, 2012.

[3] BG Shinn-Cunningham, N Kopco, and TJ Martin, “Localizing nearby sound sources in a classroom: Binaural room impulse responses”, *The Journal of the Acoustical Society of America*, vol. 117, no. 5, pp. 3100–3115, 2005.

[4] E Larsen, N Iyer, CR Lansing, and AS Feng, “On the minimum audible difference in direct-to-reverberant energy ratio”, *The Journal of the Acoustical Society of America*, vol. 124, no. 1, pp. 450–461, 2008.

[5] AJ Kolarik, BCJ Moore, P Zahorik, S Cirstea, and S Pardhan, “Auditory distance perception in humans: a review of cues, development, neuronal bases, and effects of sensory loss”, *Attention, Perception, Psychophysics*, vol. 78, no. 2, pp. 373–395, 2015.

[6] J. Blauert, J.S. Allen, and MIT Press”, *Spatial Hearing*, Amsterdam University Press, Amsterdam, Netherlands, 1997.

[7] JJ Jetzt, “Critical distance measurement of rooms from the sound energy spectral response”, *The Journal of the Acoustical Society of America*, vol. 65, no. 5, pp. 1204–1211, 1979.

[8] V Best, R Baumgartner, M Lavandier, P Majdak, and N Kopčo, “Sound Externalization: A Review of Recent Research”, *Trends in Hearing*, vol. 24, pp. 233121652094839, 2020.

[9] WM Hartmann and A Wittenberg, “On the externalization of sound images”, *The Journal of the Acoustical Society of America*, vol. 99, no. 6, pp. 3678–3688, 1996.

[10] R Baumgartner, DK Reed, B Tóth, V Best, P Majdak, HS Colburn, and B Shinn-Cunningham, “Asymmetries in behavioral and neural responses to spectral cues demonstrate the generality of auditory looming bias”, *Proceedings of the National Academy of Sciences*, vol. 114, no. 36, pp. 9743–9748, 2017.

[11] AW Boyd, WM Whitmer, JJ Soraghan, and MA Akeroyd, “Auditory externalization in hearing-impaired listeners: The effect of pinna cues and number of talkers”, *The Journal of the Acoustical Society of America*, vol. 131, no. 3, pp. EL268–EL274, 2012.

[12] U Zölzer, *DAFX: Digital Audio Effects*, Wiley, 2 edition, 2011.

[13] TR Letowski and ST Letowski, “Auditory Spatial Perception: Auditory Localization”, *Army Research Laboratory*, 2012.

[14] WO Brimijoin, AW Boyd, and MA Akeroyd, “The Contribution of Head Movement to the Externalization and Internalization of Sounds”, *PLoS ONE*, vol. 8, no. 12, pp. e83068, 2013.

[15] E Hendrickx, P Stitt, JC Messonnier, JM Lyzwa, BF Katz, and C de Boishéraud, “Influence of head tracking on the externalization of speech stimuli for non-individualized binaural synthesis”, *The Journal of the Acoustical Society of America*, vol. 141, no. 3, pp. 2011–2023, 2017.

[16] S Li, J E, R Schlieper, and J Peissig, “The impact of trajectories of head and source movements on perceived externalization of a frontal sound source”, *The Journal of the Acoustical Society of America*, may 2018.

[17] S Ji, R Schlieper, and J Peissig, “The impact of head movement on perceived externalization of a virtual sound source with different brir lengths”, *The Journal of the Acoustical Society of America*, march 2019.

[18] V Valimaki, JD Parker, L Savioja, JO Smith, and JS Abel, “Fifty Years of Artificial Reverberation”, *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 5, pp. 1421–1448, 2012.

[19] E Armelloni, C Giottoli, and A Farina, “Implementation of real-time partitioned convolution on a DSP board”, *2003 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (IEEE Cat. No.03TH8684)*, 2003.

[20] WG Gardner, “Efficient Convolution without Input-Output Delay”, *The Journal of the Acoustical Society of America*, vol. 43, no. 3, 1995.

[21] A Torger and A Farina, “Real-time partitioned convolution for ambiophonics surround sound”, in *Proceedings of the 2001 IEEE Workshop on the Applications of Signal Processing to Audio and Acoustics (Cat. No.01TH8575)*, 2001, pp. 195–198.

[22] G Kearney, C Masterson, S Adams, and F Boland, “Dynamic time warping for acoustic response interpolation: Possibilities and limitations”, in *2009 17th European Signal Processing Conference*, 2009, pp. 705–709.

[23] V Garcia-Gomez and JJ Lopez, “Binaural room impulse responses interpolation for multimedia real-time applications”, *journal of the audio engineering society*, may 2018.

[24] K Meesawat and D Hammershøi, “An Investigation on the Transition from Early Reflections to a Reverberation Tail in a BRIR”, *International Conference on Auditory Display*, 2002.

[25] A Lindau, H Maempel, and S Weinzierl, “Minimum BRIR grid resolution for dynamic binaural synthesis”, *The Journal of the Acoustical Society of America*, vol. 123, no. 5, pp. 3498–3498, 2008.

[26] W Guo, J Ren, B Wang, and Q Zhu, “Effects of Relaxing Music on Mental Fatigue Induced by a Continuous Performance Task: Behavioral and ERPs Evidence”, *PLOS ONE*, vol. 10, no. 8, pp. e0136446, 2015.

[27] A Ishii, M Tanaka, M Iwamae, C Kim, E Yamano, and Y Watanabe, “Fatigue sensation induced by the sounds associated with mental fatigue and its related neural activities: revealed by magnetoencephalography”, *Behavioral and Brain Functions*, vol. 9, no. 1, pp. 24, 2013.

[28] Y Yasunori and K Masatomo, “Detecting mental fatigue from eye-tracking data gathered while watching video: Evaluation in younger and older adults”, *Artificial Intelligence in Medicine*, vol. 91, pp. 39–48, 2018.

[29] PM Giller, F Wendt, and R Holdrich, “The influence of different BRIR modification techniques on externalization and sound quality”, pp. 61–66, 2019.

[30] J Catic, S Santurette, and T Dau, “The role of reverberation-related binaural cues in the externalization of speech”, *The Journal of the Acoustical Society of America*, vol. 138, no. 2, pp. 1154–1167, 2015.

[31] P Zahorik and Bronkhorst AW Brungart, DS, “Auditory Distance Perception in Humans: A Summary of Past and Present Research”, *Acta Acustica united with Acustica*, vol. 91, no. 3, pp. 409–420, 2005.

[32] DH Mershon and LE King, “Intensity and reverberation as factors in the auditory perception of egocentric distance”, *Perception Psychophysics*, vol. 18, no. 6, pp. 409–415, 1975.

A Appendix

A.1 FFTW

The Fastest Fourier Transform in the West[13] (FFTW) is, as the name suggests, the fastest free software implementation of the fast Fourier transform (FFT). It is a C subroutine library that allows to compute the FFT and the discrete cosine/sine transforms (DCT/DST) in one or more dimensions. Many numeric computing platform such as MATLAB and Octave, base their FFT functions on the FFTW library.

Calls to FFTW for the computation of one-dimensional DFTs of real data are structured as follows:

```
#include <fftw.h>
...
{
     in = fftw_malloc(sizeof(float) * 2 * (N/2 + 1)));
     out = (fftw_complex *) in;
     fftw_plan p;
     ...
     p = fftwf_plan_dft_r2c_1d(N, in, out, FFTW_MEASURE);
     ...
     fftw_execute(p);
     ...
     fftw_destroy_plan(p);
}
```

The input `in` is assumed to be be `N` real numbers, while the output `out` is `N/2 + 1` complex numbers (the non-redundant outputs). In this case in-place transformation in performed, requiring the input and output arrays to be of the same size, so the real array `in` has to be padded to `2 * (N/2 + 1)`. The FFTW's *planner*, upon calling the `fftwf_plan_dft_r2c_1d()` routine, produces a data structure called a *plan* that contains the information required by the library to compute the transform in the fastest way. The planner learns the optimal way of computing the FFT according to an automatic performance adaptation on the machine on which is run. The plan can be executed by calling the `fftw_execute()` routine. The plan can be called multiple times and when it is no more required it must be deallocated by calling the `fftw_destry_plan()` routine. The flag argument can be either `FFTW_MEASURE` or `FFTW_ESTIMATE`. `FFTW_MEASURE` means that FFTW actually runs and measures the execution time of several FFTs in order to find the best way to compute the transform of size `N`. `FTW_ESTIMATE`, on the contrary, does not run any computation, and just builds a reasonable plan, which may be sub-optimal.

[13] FFTW documentation: http://www.fftw.org/