METHODOLOGY

Compression of Room Impulse Responses for Compact Storage and Fast Low-latency Convolution

Martin Jälmby[1*], Filip Elvander[2] and Toon van Waterschoot[1]

Abstract

Room impulse responses (RIRs) are used in several applications, such as augmented reality and virtual reality. These applications require a large number of RIRs to be convolved with audio, under strict latency constraints. In this paper we consider the compression of RIRs, in conjunction with fast time-domain convolution. We consider three different methods of RIR approximation for the purpose of RIR compression, and compare them to state-of-the-art compression. The methods are evaluated using several standard objective quality measures, both channel-based and signal-based. We also propose a novel low-rank-based algorithm for fast time-domain convolution, and show how the convolution can be carried out without the need to decompress the RIR. Numerical simulations are performed using RIRs of different lengths, recorded in three different rooms. It is shown that compression using low-rank approximation is a very compelling option to the state-of-the-art Opus compression, as it performs as well or better than on all but one considered measure, with the added benefit of being amenable to fast time-domain convolution.

Keywords: Low-rank modeling; room impulse responses; convolution; tensor decomposition

1 Introduction

Modeling the acoustics of a room as a linear time-invariant system, the room impulse response (RIR) describes the impact of the room on an acoustic excitation signal, from a certain source position to a certain receiver position. The availability of the RIR, or an accurate estimate thereof, is imperative to a multitude of acoustic signal processing tasks, such as source localization [1], speech dereverberation [2], auralization [3, 4], source separation [5], listening room compensation [6], and echo cancellation [7]. There are several ways of modeling the RIR. Among the more popular ones are the infinite impulse response (IIR) (see e.g., [8, 9]) and finite impulse response (FIR) (see e.g., [8, 10]) models. The IIR model offers the possibility of a more compact representation, however with the downside of possible difficulties estimating the filter parameters [11], and potential issues with instability [12]. The FIR model is simple and straightforward, but with the disadvantage that comparatively many coefficients are needed to accurately represent the RIR [11]. For example, for an office-sized room, the FIR model can be several thousands of taps long [2]. A concert hall, on the other hand, can have a reverberation time of a couple of seconds [13], which at a sampling rate of 48 kHz yields an RIR with a length on the order of 10^5 samples. This can be prohibitive from both a memory requirement and computational complexity point for view, when using the RIR for convolution [14–16].

In recent years, archaeoacoustics and the cultural heritage preservation of acoustic scenes has gained increased interest from the research community, see e.g., [17] and references therein. However, in order to faithfully reconstruct the sound field in a room, the spatial resolution of the grid of measurements needs to be on the order of 10 cm [18]. Considering that the RIR depends on both the source and receiver position, even for a small room, the number of required source/receiver configurations for which the RIR has to be measured and stored will be in the millions, hence amounting to hundreds of gigabytes of data for the acoustic representation of a single room, indicating a significant need for compact representations of RIRs.

The computational challenges posed by long RIRs are particularly apparent in acoustic signal processing applications requiring low input-output latency, such

*Correspondence: martin.jalmby@esat.kuleuven.be
[1]Department of Electrical Engineering (ESAT/STADIUS), KU Leuven, Leuven, Belgium
Full list of author information is available at the end of the article

as virtual video conferencing [19], augmented/mixed reality [20] and virtual reality [3, 21], sound zone control [22, 23], network music performance [24], and artificial reverberation [25]. In this paper, we consider rendering techniques based on convolution, highlighting the need for fast, low-latency convolution with long RIRs.

Ever since the works of Cooley and Tukey [26], and Stockham [27], a popular approach has been to carry out convolution in the frequency domain. With the help of the convolution theorem, which states that (circular) convolution in the (discrete) time domain is equivalent to point-wise multiplication in the (discrete) frequency domain, one is able to significantly reduce the computational burden of convolution in most cases, owing to the computational efficiency of the fast Fourier transform (FFT) algorithm. Frequency-domain convolution has since been further improved by methods such as overlap-add (OLA) and overlap-save (OLS), and partitioned convolution. For an overview of these methods see e.g., [28,29]. A drawback of frequency-domain convolution is, however, that it is block-based, and therefore inevitably introduces latency. Further, partitioned frequency-domain filters are subject to restrictions with regards to assembling them into networks of filters (in parallel or serial structure), which is not the case for time-domain filters [30]. Another possible way to attempt to speed up the computations is by perceptual convolution [31]. There, the convolution is simplified, based on a perceptual criterion. The number of frequency-domain multiplications, and the memory storage, are reduced by up to 60%, without considerable quality degradation. Another path is optimization with respect to processor architecture, and the use of graphics processing units (see e.g., [32] and references therein). Yet another approach is to effectively shorten the RIR by treating the different parts of the RIR separately. For example, in [33], convolution is carried out for the first parts of the RIR, corresponding to the direct component and early reflections. The late reverberation, however, is modeled as a velvet noise sequence, yielding a very sparse FIR filter. Instead of being convolved with the sparse FIR, the input signal is propagated in the delay line of the filter, and only the samples coinciding with a non-zero component of the sparse FIR are added together to yield the output.

In this paper we consider RIR compression and fast low-latency time-domain convolution based on three different methods; truncation, (hard) thresholding, and low-rank approximation. The exploitation of the (approximate) low-rank structure of reshaped RIRs is something we have considered in previous work. The physical motivation for it, and its applicability to real-life RIRs, was demonstrated in [34]. How the low-rank structure can be exploited when estimating RIRs from noisy input-output relations was shown in [35] and the simultaneous compression of multiple RIRs was considered in [36]. Atkins *et al.* showed in [37] how this low-rank structure can be exploited in time-domain convolution, an idea we extended upon in recent work [38]. Jaderberg *et al.* showed in [39] how speeding up convolutional neural networks can be done by leveraging low rank, but the authors consider dimensions no higher than 3.

The contribution of this paper is threefold. Firstly, we provide an extensive comparison of the aforementioned compression methods, with respect to several objective quality measures, both channel-based and signal-based. Secondly, we propose an approximate fast time-domain convolution method based on N-D low-rank tensor approximation of an RIR. This yields lower computational complexity than traditional time-domain convolution, and lower latency than FFT-based fast convolution. Thirdly, we show how the problem of compression and fast time-domain convolution can be handled within the same framework. This comes with the major advantage that the compressed RIR does not need to be decompressed before it can be used for convolution.

This paper is organized as follows: first, Section 1 is concluded with an introduction of the notation used throughout the paper, as well as the introduction of the signal model. In Section 2, the different RIR approximations considered for RIR compression are introduced. In Section 3, convolution by low-rank approximation is introduced. Section 4 introduces the different objective quality measures that will be used for evaluation. Numerical results are presented in Section 5, and finally, conclusions are presented in Section 6.

1.1 Notation And Signal Model

We denote scalars, vectors, matrices, and tensors by lowercase (e.g., h), bold lowercase (e.g., $\mathbf{h}$), bold uppercase (e.g., $\mathbf{H}$), and calligraphic letters (e.g., $\mathcal{H}$), respectively. Sets are also denoted by calligraphic letters, but it will be clear from context what is considered. The selection of one or several elements from a vector, matrix, or tensor will be denoted by square brackets, e.g. $\mathbf{H}[m:n, j]$ is a vector containing the mth till the nth element of the jth column of $\mathbf{H}$, and the hat symbol, $\hat{\cdot}$, indicates an approximated quantity. The symbol $\circ$ denotes the outer product, i.e., $(\mathbf{x}_1 \circ \mathbf{x}_2 \circ \cdots \circ \mathbf{x}_D)[j_1, j_2, \ldots, j_D] = \mathbf{x}_1[j_1]\mathbf{x}_2[j_2]\ldots\mathbf{x}_D[j_D]$, $(:)$ denotes vectorization of a matrix or a tensor, and $\lfloor\cdot\rfloor$ denotes the flooring operation.

We consider a discrete-time RIR $h(k)$, for $k = 0, 1, ..., n_h - 1$, arranged in a vector $\mathbf{h} \in \mathbb{R}^{n_h}$, as well

as a discrete-time signal $x(k)$, for $k = 1, 2, \ldots, n_x$, arranged in the vector $\mathbf{x} \in \mathbb{R}^{n_x}$. The convolution of these vectors yields the discrete-time output

$$y(k) = \sum_{n=0}^{n_h-1} h(n)x(k-n), \tag{1}$$

for $k = 1, 2, \ldots, n_y$, with corresponding vector $\mathbf{y} \in \mathbb{R}^{n_y}$, where $n_y = n_h + n_x - 1$. Generally, throughout this paper, an element is considered to be 0, if the index is out of its defined range, equivalent to appropriate zero-padding.

2 Room Impulse Response Compression

We will consider three different RIR approximations for RIR compression, and compare them to a state-of-the-art compression benchmark.

2.1 Compression by truncation

Firstly, we consider an RIR compressed by *truncation*, $\hat{\mathbf{h}}_{\mathrm{T}}$, where

$$\hat{\mathbf{h}}_{\mathrm{T}}(n) = \begin{cases} \mathbf{h}(n), n \leq n_{\mathrm{T}} \\ 0, n > n_{\mathrm{T}} \end{cases} \tag{2}$$

for some $n_{\mathrm{T}} \in \mathbb{N}, n_{\mathrm{T}} \leq n_h$. This method is amenable to accelerated convolution, as the length of the impulse response is shortened, decreasing the number of multiply-add instructions per output sample from n_h to n_{T}.

2.2 Compression by thresholding

Secondly, we consider an RIR compressed by *thresholding* [1], $\hat{\mathbf{h}}_{\mathrm{K}}$, defined as

$$\hat{\mathbf{h}}_{\mathrm{K}}(n) = \begin{cases} \mathbf{h}(n), n \in \mathcal{K}_{n_{\mathrm{k}}} \\ 0, n \notin \mathcal{K}_{n_{\mathrm{k}}} \end{cases} \tag{3}$$

where $\mathcal{K}_{n_{\mathrm{k}}}$ is the set of indices of the n_{k}, in absolute value, largest elements of $\mathbf{h}$. Also this RIR approximation method yields a possibly faster convolution. As many of the elements of $\hat{\mathbf{h}}_{\mathrm{K}}$ are zero, these do not have to be considered in the convolution. For a sparse impulse response $\hat{\mathbf{h}}_{\mathrm{K}}$ we can define the convolution between $\hat{\mathbf{h}}_{\mathrm{K}} \in \mathbb{R}^{n_h}$ and $\mathbf{x} \in \mathbb{R}^{n_x}$ as

$$y(k) = \sum_{n \in \mathcal{K}_{n_{\mathrm{k}}}} \hat{\mathbf{h}}_{\mathrm{K}}(n)x(k-n). \tag{4}$$

This reduces the number of multiply-add instructions per output sample from n_h to n_{k}. The argument could be made that the positions of the non-zero components need to be stored, and that that is something that needs to be taken into account as well. However, whereas the coefficients themselves are floating numbers, the positions are integers, taking up significantly less space. Therefore, the impact of having to store the positions was ignored when considering the compression of thresholding.

[1] In previous work, [34], we have referred to this as *KMax*.

2.3 Compression by low-rank approximation

Lastly, we consider an RIR compressed by *low-rank* approximation, $\hat{\mathbf{h}}_{\mathrm{LR}}$. Assuming $n_h = n_{s_1} n_{s_2}$, with $n_{s_1}, n_{s_2} \in \mathbb{N}$, the RIR $\mathbf{h} \in \mathbb{R}^{n_h}$ can be reshaped into a matrix $\mathbf{H} \in \mathbb{R}^{n_{s_1} \times n_{s_2}}$,

$$\mathbf{H} = \begin{bmatrix} h(1) & h(n_{s_1}+1) & \ldots & h(n_{s_1}(n_{s_2}-1)+1) \\ \vdots & \vdots & & \vdots \\ h(n_{s_1}) & h(2n_{s_1}) & \ldots & h(n_h) \end{bmatrix}. \tag{5}$$

With the use of the singular value decomposition (SVD) $\mathbf{H} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$ and assuming the singular values in $\mathbf{\Sigma}$ are arranged in non-increasing order, we can then make a rank-R approximation of $\mathbf{H}$,

$$\hat{\mathbf{H}}_{\mathrm{2D}} = \mathbf{U}[:, 1:R]\mathbf{\Sigma}[1:R, 1:R]\mathbf{V}[:, 1:R]^T. \tag{6}$$

Finally, $\hat{\mathbf{h}}_{\mathrm{2D}} = \hat{\mathbf{H}}_{\mathrm{2D}}(:)$. Similarly, assuming $n_h = \prod_{d=1}^{D} n_{s_d}$, $n_{s_d} \in \mathbb{N}$, the vector $\mathbf{h}$ can be reshaped into a tensor $\mathcal{H} \in \mathbb{R}^{n_{s_1} \times n_{s_2} \times \cdots \times n_{s_D}}$, of arbitrary dimension D, where n_{s_d} denotes the size of the dth dimension and the rank of a tensor is defined as the smallest number of rank-1 tensors needed to generate the tensor $\mathcal{H}$ as their sum. In a similar fashion as to the matrix, we can then make a rank-R approximation $\hat{\mathcal{H}}_{\mathrm{LR}}$ of $\mathcal{H}$. For this we will be using a (canonic) polyadic decomposition (see e.g., [40] and references therein). This is done using the high-level function *cpd* of the Matlab toolbox Tensorlab [41]. Subsequently, $\hat{\mathbf{h}}_{\mathrm{LR}} = \hat{\mathcal{H}}_{\mathrm{LR}}(:)$. We will, in addition to aforementioned $\hat{\mathbf{h}}_{\mathrm{2D}}$, consider low-rank approximation of 3-D and 5-D tensors, denoted $\hat{\mathbf{h}}_{\mathrm{3D}} = \hat{\mathcal{H}}_{\mathrm{3D}}(:)$ and $\hat{\mathbf{h}}_{\mathrm{5D}} = \hat{\mathcal{H}}_{\mathrm{5D}}(:)$, respectively. The absence of a 4-D tensor approximation is explained in Section 5. Also the low-rank approximation method allows for fast time-domain convolution, which we have explored in recent work for up to three dimensions [38]. Here we will extend this idea to tensors of arbitrary dimensions. This will be further explained in Section 3.

2.4 Compression benchmark: Opus

The three methods proposed above, truncation, thresholding, and low-rank approximation, will be compared

to the state-of-the-art Opus interactive speech and audio codec [42,43]. The Opus codes is created from two core technologies: Skype's SILK codec [44], based on Linear Prediction (LP), and Xiph.Org's CELT codec [45,46], based on the Modified Discrete Cosine Transform (MDCT). The idea behind this construction is that LP is considered to code low frequencies more efficiently, whereas for music and higher speech frequencies, MDCT is superior. The double layers allow Opus to achieve higher quality for a wide range of audio. The Opus codec was created for, and has previously mainly been considered for, speech and music, but it has recently also gained attention as a possible way to compress RIRs [47]. In this work, the Opus encoding was done using Matlab's *audiowrite*. It should be noted that although Opus shrinks the file size of the stored RIR, the number of coefficients remains the same. The RIR compressed by Opus, that will be denoted $\hat{\mathbf{h}}_{\mathrm{O}}$, is therefore, to the best of the authors' knowledge, not amenable to fast time-domain convolution. In order to give the reader a feel for the different approximations, an example RIR, taken from [48], and a selection of the compressed RIRs obtained with the different compression methods, at a compression rate (see (32)) of 0.8, are displayed in Fig. 1.

3 Convolution By Low-rank Approximation

Accelerating convolution by exploiting low-rank approximations was first considered by Atkins *et. al* in [37]. The authors there considered a low-rank approximation of a matricization of the RIR, using the SVD. In recent work, [38], we have extended this idea to a three-dimensional tensorization of the RIR. We will here show how this idea can be further extended to a tensorization of arbitrary dimension. We will first demonstrate the 2-D case presented in [35], and then explain the extension to a tensor of arbitrary dimension.

3.1 Partitioned Truncated SVD Filter

Assuming $n_h = n_{s_1} n_{s_2}$, for $n_{s_1}, n_{s_2} \in \mathbb{N}$, an output sample $y(k)$ of the convolution in (1) can be written as

$$y(k) = \sum_{j=1}^{n_{s_2}} \mathbf{x}_k^{(j)^T} \mathbf{h}^{(j)}, \tag{7}$$

where

$$\mathbf{h}^{(j)} \triangleq \begin{bmatrix} h((j-1)n_{s_1}) & \dots & h(jn_{s_1}-1) \end{bmatrix} \in \mathbb{R}^{n_{s_1}} \tag{8}$$

and

$$\mathbf{x}_k^{(j)} \triangleq \begin{bmatrix} x(k-(j-1)n_{s_1}) \dots x(k-jn_{s_1}+1) \end{bmatrix} \in \mathbb{R}^{n_{s_1}}, \tag{9}$$

for $j = 1, 2, \dots, n_{s_2}$. Instead of as in (1) writing $y(k)$ as the inner product of two vectors of length $n_h = n_{s_1} n_{s_2}$, it is in (7) written as the sum of n_{s_2} inner products of vectors of length n_{s_1}. Further, the RIR $\mathbf{h}$ can be reshaped into a matrix $\mathbf{H} = \begin{bmatrix} \mathbf{h}^{(1)} & \dots & \mathbf{h}^{(n_{s_2})} \end{bmatrix} \in \mathbb{R}^{n_{s_1} \times n_{s_2}}$. For now we are going to assume that this matrix is rank-1, i.e., it can be written as the outer product $\mathbf{H} = \mathbf{s}_1 \circ \mathbf{s}_2$, for two vectors $\mathbf{s}_1 \in \mathbb{R}^{n_{s_1}}$ and $\mathbf{s}_2 \in \mathbb{R}^{n_{s_2}}$. Under this assumption, we have that

$$\mathbf{H} = \begin{bmatrix} \mathbf{s}_1 \mathbf{s}_2[1] & \mathbf{s}_1 \mathbf{s}_2[2] & \dots & \mathbf{s}_1 \mathbf{s}_2[n_{s_2}] \end{bmatrix}, \tag{10}$$

i.e., the jth column of $\mathbf{H}$, corresponding to $\mathbf{h}^{(j)}$, is the vector $\mathbf{s}_1$ scaled by $\mathbf{s}_2[j]$, $j = 1, 2, \dots, n_{s_2}$. Further, the following property is readily verified,

$$\mathbf{x}_k^{(j)} = \mathbf{x}_{k+an_{s_1}}^{(j+a)}, a \in \mathbb{Z}. \tag{11}$$

Because of (10) and (11), only the first inner product of the sum in (7) has to be computed per output sample k, the other inner products of the sum, i.e., $\mathbf{x}_k^{(j)^T} \mathbf{s}_1 = \mathbf{x}_{k-n_{s_1}}^{(j-1)^T} \mathbf{s}_1$, for $j = 2, \dots, n_{s_2}$, have already been computed for a previous time sample, and can therefore be fetched from memory and multiplied with the appropriate entry from $\mathbf{s}_2$,

$$y(k) = \left(\mathbf{x}_k^{(1)^T} \mathbf{s}_1 \right) \mathbf{s}_2[1] + \sum_{j=2}^{n_{s_2}} \underbrace{\left(\mathbf{x}_k^{(j)^T} \mathbf{s}_1 \right)}_{\text{Fetch from memory}} \mathbf{s}_2[j]. \tag{12}$$

This reduces the number of multiplications per sample to be carried out, from $n_h = n_{s_1} n_{s_2}$ to $n_{s_1} + n_{s_2}$. These ideas can be extended to a matrix $\mathbf{H}$ of arbitrary rank R. Instead of $\mathbf{H}$ being just the outer product of two vectors, it is now a sum of R outer products,

$$\mathbf{H} = \mathbf{S}_1 \mathbf{S}_2^T = \sum_{r=1}^{R} \mathbf{S}_1[:, r] \circ \mathbf{S}_2[:, r] = \sum_{r=1}^{R} \mathbf{S}_1[:, r] \mathbf{S}_2[:, r]^T, \tag{13}$$

for $\mathbf{S}_1 \in \mathbb{R}^{n_{s_1} \times R}$, and $\mathbf{S}_2 \in \mathbb{R}^{n_{s_2} \times R}$. Equation (12) can now be extended to

$$y(k) = \sum_{r=1}^{R} \left(\left(\mathbf{x}_k^{(1)^T} \mathbf{S}_1[:, r] \right) \mathbf{S}_2[1, r] + \sum_{j=2}^{n_{s_2}} \underbrace{\left(\mathbf{x}_k^{(j)^T} \mathbf{S}_1[:, r] \right)}_{\text{Fetch from memory}} \mathbf{S}_2[j, r] \right) \tag{14}$$

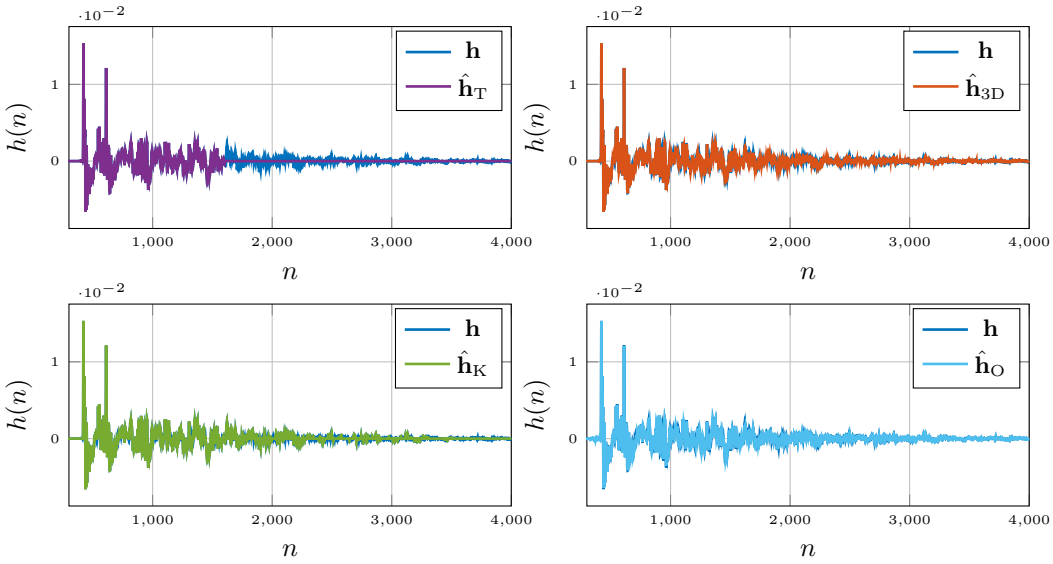

Figure 1: Example results of RIR compression methods

where only R inner products have to be computed for each time sample. Similar to (12), this reduces the number of multiplications to $R(n_{s_1} + n_{s_2})$.

3.2 Fast Time-domain Convolution by Tensor Approximation

We are now ready to extend these ideas to a tensor of arbitrary dimension. Assuming $\mathbf{h} \in \mathbb{R}^{n_h}$, with $n_h = \prod_{d=1}^{D} n_{s_d}$, for $n_{s_1}, n_{s_2}, \ldots, n_{s_D} \in \mathbb{N}$, let $\mathbf{h}$ be reshaped into a tensor $\mathcal{H} \in \mathbb{R}^{n_{s_1} \times n_{s_2} \times \cdots \times n_{s_D}}$, and assume that $\mathcal{H}$ is of rank R. Then, analogously to (13),

$$\mathcal{H} = \sum_{r=1}^{R} \mathbf{S}_1[:, r] \circ \mathbf{S}_2[:, r] \circ \cdots \circ \mathbf{S}_D[:, r], \tag{15}$$

where $\mathbf{S}_d \in \mathbb{R}^{n_{s_d} \times R}$, $d = 1, 2, \ldots, D$, and in analog to (10), but with arbitrary dimension and rank, we have that

$$\mathcal{H}[:, j_2, j_3, \ldots, j_D] = \sum_{r=1}^{R} \mathbf{S}_1[:, r]\mathbf{S}_2[j_2, r] \ldots \mathbf{S}_D[j_D, r]. \tag{16}$$

The equality of (11) can be generalized according to

$$\mathbf{x}_k^{(j_2, j_3, \ldots, j_D)} = \mathbf{x}_{k + \sum_{d=2}^{D} a_d \prod_{p=1}^{d-1} n_{s_p}}^{(j_2 + a_2, j_3 + a_3, \ldots, j_D + a_D)}, \tag{17}$$

where $\mathbf{x}_k^{(j_2, j_3, \ldots, j_D)} \in \mathbb{R}^{n_{s_1}}$ is a vector containing the n_{s_1} latest samples of $\mathbf{x}$, in reversed order, starting at $x(k - \sum_{d=2}^{D}(j_d - 1)\prod_{p=1}^{d-1} n_{s_p})$, and $a_2, a_3, \ldots, a_D \in \mathbb{Z}$. While verifying (17) can seem like a daunting task, it becomes clearer when considering the indices of the first entry of the vectors on the left and right hand side of (17), respectively,

$$\begin{aligned} &k - \sum_{d=2}^{D}(j_d - 1)\prod_{p=1}^{d-1} n_{s_p} = \\ &k + \sum_{d=2}^{D} a_d \prod_{p=1}^{d-1} n_{s_p} - \sum_{d=2}^{D}(j_d + a_d - 1)\prod_{p=1}^{d-1} n_{s_p}. \end{aligned} \tag{18}$$

The pattern from (7) extends to

$$y(k) = \sum_{j_2=1}^{n_{s_2}} \cdots \sum_{j_D=1}^{n_{s_D}} \mathbf{x}_k^{(j_2, j_3, \ldots, j_D)^T} \mathbf{h}^{(j_2, j_3, \ldots, j_D)}, \tag{19}$$

where $\mathbf{h}^{(j_2, j_3, \ldots, j_D)} = \mathcal{H}[:, j_2, j_3, \ldots, j_D]$ is a vector containing n_{s_1} consecutive elements of $\mathbf{h}$, starting at $h(\sum_{d=2}^{D}(j_d - 1)\prod_{p=1}^{d-1} n_{s_p})$. Subsequently, the property of (14) is generalized to

$$\begin{aligned} y(k) = \sum_{r=1}^{R} \Bigg(&\mathbf{x}_k^{(1, \ldots, 1)^T} \mathbf{S}_1[:, r]\mathbf{S}_2[1, r] \ldots \mathbf{S}_D[1, r] + \\ &\sum_{j_2=2}^{n_{s_2}} \cdots \sum_{j_D=2}^{n_{s_D}} \underbrace{\left(\mathbf{x}_k^{(j_2, \ldots, j_D)^T} \mathbf{S}_1[:, r]\right)}_{\text{Fetch from memory}} \mathbf{S}_2[j_2, r] \ldots \mathbf{S}_D[j_D, r] \Bigg) \end{aligned} \tag{20}$$

with a corresponding structure of what has to be computed and what can be fetched from memory. Similarly

to the previous case, we have a reduction in complexity. Only R inner products of length n_{s_1} have to be computed for each time index k, reducing the number of multiplications to $R\sum_{d=1}^{D} n_{s_d}$. When naively implemented, the sum in (20) will yield many superfluous operations, where one of the vectors contains only zeros. To fully exploit the structure of the RIR, and to maximize efficiency, it is therefore important to keep track of which operations actually need to be carried out and keep the number of multiplications with zeros to a minimum. We here propose an explicit algorithm.

Let $\mathcal{H} = \sum_{r=1}^{R} \mathbf{S}_1[:,r] \circ \mathbf{S}_2[:,r] \circ \cdots \circ \mathbf{S}_D[:,r]$, where $\mathcal{H} \in \mathbb{R}^{n_{s_1} \times n_{s_2} \times \cdots \times n_{s_D}}$, and $\mathbf{S}_d \in \mathbb{R}^{n_{s_d} \times R}$, for $d = 1, 2, \ldots, D$. The operator $\mathcal{I} : \mathbb{R}^n \to \mathbb{R}^n$ denotes the reversion of the order of the elements in a vector, i.e., $\mathcal{I}(\mathbf{x}) = \begin{bmatrix} x(n_x) & x(n_x - 1) \ldots & x(1) \end{bmatrix}^T$, and $\mathbf{0}_n \in \mathbb{R}^n$ is a vector of zeros. The foundation of the algorithm is that, for each k, we compute the R necessary inner products, store the resulting values to memory and add these to $y(k)$ with appropriate scaling by the corresponding elements of $\mathcal{H}$. Next, the remaining non-zero inner products in the sum of (20) are fetched from memory, scaled by the corresponding entry of $\mathcal{H}$ and added to $y(k)$. The fast low-latency convolution algorithm by low-rank tensor approximation is summarized in Algorithm 1. A few remarks regarding Algorithm 1, for providing intuition as well as clarity, are in order:

- New inner products need to be computed and stored to memory as long as $k \leq n_{s_1} + n_x - 1$, this is done within the if-statement starting at line 5.
- Within the for-statement starting at line 14 the old inner products are fetched from memory and added to the output.
- On line 15, for $d = 2$, the upper limit of $\prod_{p=2}^{d-1} n_{s_p}$ is lower than the lower limit, in which case, by convention, $\prod_{p=2}^{1} n_{s_p} = 1$.

Algorithm 1: Fast Low-latency Convolution by Low-rank Tensor Approximation

1 Input: $\mathcal{H} = \sum_{r=1}^{R} \mathbf{S}_1[:,r] \circ \mathbf{S}_2[:,r] \circ \ldots \circ \mathbf{S}_D[:,r]$, $\mathbf{x}$
2 Output: $\mathbf{y}$
3 **for** $k = 1, 2, \ldots, n_y$ **do**
4 **for** $r = 1, 2, \ldots, R$ **do**
5 **if** $k \leq n_{s_1} + n_x - 1$ **then**
6 $u_{\mathsf{b}} = \mathsf{max}(k - n_x + 1, 0)$;
7 $u_{\mathsf{e}} = \mathsf{min}(k, n_{s_1})$;
8 $x_{\mathsf{b}} = \mathsf{max}(k - n_{s_1} + 1, 1)$;
9 $x_{\mathsf{e}} = \mathsf{min}(k, n_x)$;
10 $\mathbf{C}[\mathsf{mod}(k-1, n_h) + 1, r] = \left[\mathcal{I}(\mathbf{x}[x_{\mathsf{b}} : x_{\mathsf{e}}])\right]^T \mathbf{S}_1[u_{\mathsf{b}} : u_{\mathsf{e}}, r]$;
11 $y(k) = \prod_{d=2}^{D} \mathbf{S}_d[1,r]\mathbf{C}[\mathsf{mod}(k-1,n_h)+1,r]$;
12 $l = \mathsf{max}\left(\lfloor (k - n_x)/n_{s_1} \rfloor + 1, 2\right)$;
13 $u = \mathsf{min}\left(\lfloor (k-1)/n_{s_1} \rfloor + 1, \prod_{d=2}^{D} n_{s_d}\right)$;
14 **for** $c = l, l+1, \ldots, u$ **do**
15 $j_d = \left\lfloor \mathsf{mod}\left((c-1), \prod_{p=2}^{d} n_{s_p}\right) \Big/ \prod_{p=2}^{d-1} n_{s_p} \right\rfloor + 1$; $d = 2, \ldots, D$
16 $\tilde{c} = k - \sum_{d=2}^{D} (j_d - 1) \prod_{p=1}^{d-1} n_{s_p}$
17 $y(k) = y(k) + \prod_{d=2}^{D} \mathbf{S}_d[j_d, r]\mathbf{C}[\mathsf{mod}(\tilde{c} - 1, n_h), r]$;

3.3 Complexity

By the authors of [37], it was noted that an output sample $y(k)$ requires $R(n_{s_1} + n_{s_2})$ multiply-add instructions, in the two-dimensional case, compared to the $n_h = n_{s_1} n_{s_2}$ multiply-add instructins of conventional FIR filter convolution. The computational complexity for a general, D-dimensional tensorization is a generalization of the one in [37], and amounts to $R\sum_{d=1}^{D} n_{s_d}$ multiply-add instructions, as compared to $n_h = \prod_{d=1}^{D} n_{s_d}$ multiply-add instructions of conventional FIR filter convolution. Further, as the contribution to the end result of the entries in the sum of (20) are independent from each other, it is possible to perform these computations in parallel. To provide some intuition, an example is shown in Fig. 2. Here the complexity of traditional time-domain convolution is, for varying values of n_h, compared to that of the proposed algorithm for the case of square 2-D matricization and 3-D tensorizations of rank 4 and 12.

The two-dimensional algorithm from [37] requires a memory of size $R(n_{s_1} + n_{s_2} + n_h) + n_{s_1}$ variables, compared to $2n_h$ for a conventional FIR filter. For the proposed method, it is $R(\sum_{d=1}^{D} n_{s_d} + n_h) + n_{s_1}$, i.e., also the memory requirement for the proposed method is a generalization of the one in [37].

4 Objective Quality Measures

Audio technology can generally be designed to be either physically motivated or perceptually motivated. Physically motivated techniques are typically computationally intensive, in the attempt to physically accurately represent the sound field. Perceptually motivated systems are in general less computationally demanding, as they aim only to be accurate enough for human perception [16]. The physical accuracy of low-rank approximations of RIRs was evaluated in [34], in this work we aim to investigate the perceptual accuracy of compression by low-rank approximation and the other aforementioned compression methods. In this

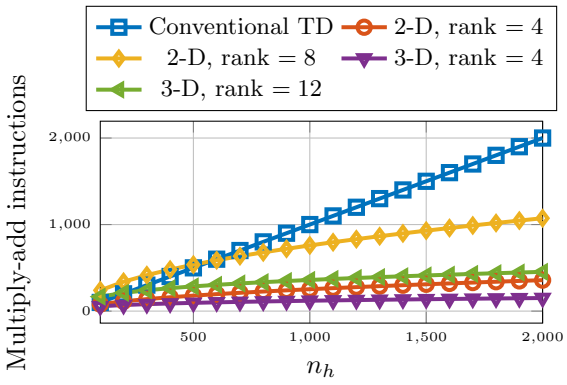

Figure 2: Computational complexity of proposed convolution

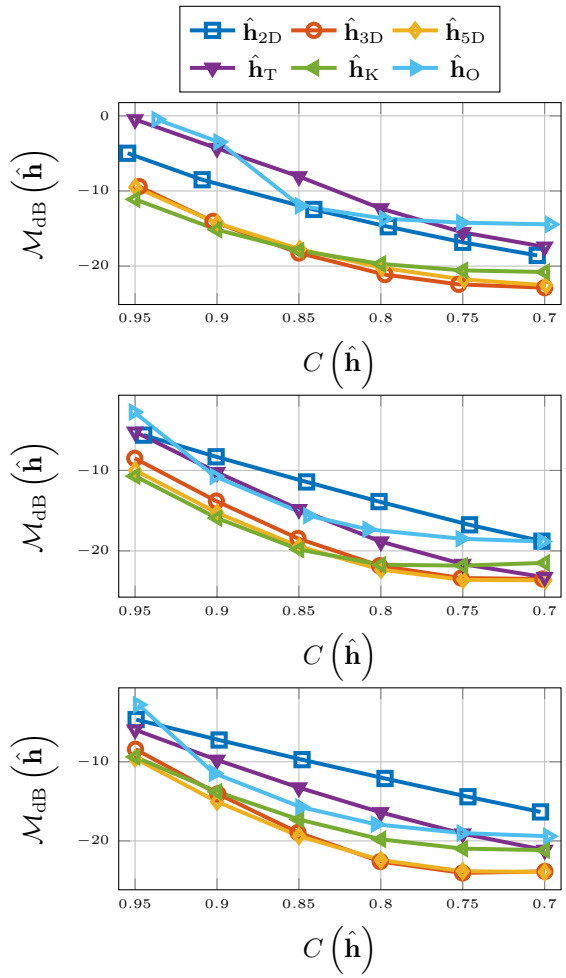

Figure 3: Normalized misalignment for SMARD (top), AIL (middle), and SAL (bottom)

section we describe a variety of parameters regarding the perception of room acoustics and corresponding objective measures. These measures can be divided into two categories, channel-based objective measures and signal-based objective measures [2]. The channel-based measures concern only how well the approximation of the channel, i.e., the compressed RIR, relates to the measured channel, i.e., the RIR. Signal-based measures, on the other hand, pertain to how the approximated channel distorts the signal output, after the compressed RIR has been convolved with e.g., music or speech.

The objective of the different measures considered here differs slightly. For some of them a high value is desirable, for others a lower value is better. For most of them, however, invariance is what is sought after, i.e. that the value of the measured quantity for a compressed RIR is as close as possible to the measured quantity for the original RIR. For an easy overview for the reader, the measures considered in this paper, their definitions, whether they are channel- or signal-based, and their objectives, are recapped in Table 1.

4.1 Channel-based Objective Quality Measures

The perhaps most obvious way to measure the quality of a compressed RIR is by the *normalized misalignment*, defined as

$$\mathcal{M}_{\mathrm{dB}}\left(\hat{\mathbf{h}}\right) = 20\log_{10}\left(\frac{\|\hat{\mathbf{h}}-\mathbf{h}\|_2}{\|\mathbf{h}\|_2}\right). \tag{21}$$

The problem with this measure is, however, that it is not necessarily a good indicator of whether the compressed RIR will yield an auditory perception faithful to the original RIR.

Reverberation time is a well-known objective measure for room acoustics. This is the time it takes for the sound level to drop 60 dB, after a stationary sound source has been switched off, and is denoted T_{60}. In practice, this measure is typically estimated as double the time it takes for the sound level to drop from −5 dB to −35 dB [13]. Finding the time it takes for the sound level to drop a certain amount is done via the energy decay curve which, since the work by Schroeder [52], is most commonly found using backwards integration. As we consider discrete-time signals in this paper, the energy decay curve $D(n)$ is found using backwards summation,

$$D(n) = \sum_{k=n}^{n_h} h^2(k) = \sum_{k=0}^{n_h} h^2(k) - \sum_{k=0}^{n} h^2(k). \tag{22}$$

Letting $n_{-x\mathrm{dB}}$ denote the time sample when the energy decay curve $D(n)$ has decreased to x dB below

Table 1: Measures

Measure	Definition	Channel/Signal	Objective
Normalized Misalignment	(21)	Channel	Low
Reverberation Time T_{60}	(23)	Channel	Invariance
Echo Density	(24)	Channel	Invariance
Early Decay Time (EDT)	(26)	Channel	Invariance
Center Time	(27)	Channel	Invariance
TOA of Direct Component	(28)	Channel	Invariance
Frequency-weighted Log-spectral Signal Distortion (SD)	(31)	Signal	Low
ViSQOLAudio	[49–51]	Signal	High

its starting value, T_{60} is found as

$$T_{60} = 2\left(n_{-35\text{dB}} - n_{-5\text{dB}}\right) f_s, \tag{23}$$

where f_s denotes sampling frequency. Reverberation can cause degraded speech intelligibility, but it is also what gives music fullness, by blending the sounds of different instruments and voices [13]. It further provides, together with the energy ratio between direct and reverberant sound and the time of arrival of the early reflections, information about the size of a space and the distance to the boundaries [53].

The *echo density* profile of an RIR is the fraction of impulse response coefficients which lie outside the standard deviation of the coefficient amplitudes, for a particular time window. A simple and robust measure for echo density was introduced by Abel *et. al* in [54],

$$\eta(n) = \frac{1/\text{erfc}\left(1/\sqrt{2}\right)}{2\delta+1} \sum_{k=n-\delta}^{n+\delta} w(k)\mathbf{1}_{\{|h(k)|>\sigma\}}, \tag{24}$$

where $\text{erfc}\left(1/\sqrt{2}\right) = 0.3173$, $2\delta+1$ is the window length in samples, $\mathbf{1}_{\{\cdot\}}$ is an indicator function, $w(k)$ is a window function, for which $\sum_k w(k) = 1$, and

$$\sigma = \left[\sum_{k=n-\delta}^{n+\delta} w(k)h^2(k)\right]^{1/2}. \tag{25}$$

Throughout this paper we will use a Hanning window with $\delta = 550$, when $f_s = 44.1$ kHz and $\delta = 600$ when $f_s = 48$ kHz, corresponding to a window length of 25 ms, as per the discussion in [54]. Further, we will only consider the part of the echo density profile where the entire window fits.

In reverberant music or speech, later parts of the reverberation tend to be masked by the direct and early components of the next note or syllable. Therefore, the alternative measure *early decay time* (EDT), has proved to be better correlated with reverberance, a perceptual attribute of reverberation, than reverberation time, in the aforementioned scenarios [13]. The EDT is defined as

$$\text{EDT} = 6(n_{-10\text{dB}})f_s. \tag{26}$$

The parameter *center time* , denoted t_s, describes the balance between early and late energy in the RIR [13], defined as

$$t_s = \frac{\sum_{k=0}^{n_h} kh^2(k)}{\sum_{k=0}^{n_h} h^2(k)}, \tag{27}$$

i.e., the center of gravity of the RIR. Two other measures that are commonly mentioned in this context are *mode density* [55, 56] and *reflections density* [16, 57]. These are, however, better suited to characterize synthetically generated RIRs. As we here consider only real-life RIRs, these measures will not be considered in this paper.

The time of arrival (TOA) of the direct component, defined as

$$\text{TOA} = \left(\arg\max_n |h(n)|\right) \Big/ f_s, \tag{28}$$

is crucial in tasks such as room geometry estimation [58] and acoustic source localization [59]. How the TOA of the direct component is preserved by a compression method is not well captured by the normalized misalignment and will therefore be considered as a separate measure in Section 5.

For all the channel-based measures introduced above, expect normalized misalignment, we aim for a minimal deviation between the compressed and original RIR measure. We will therefore, in Section 5, present the root-mean-square error (RMSE) for these quantities,

$$\text{RMSE}_g\left(\hat{\mathbf{h}}\right) = \sqrt{\sum_{j=1}^{n_{\text{RIR}}} \left|g\left(\mathbf{h}_j\right) - g(\hat{\mathbf{h}}_j)\right|^2 \Big/ n_{\text{RIR}}}, \tag{29}$$

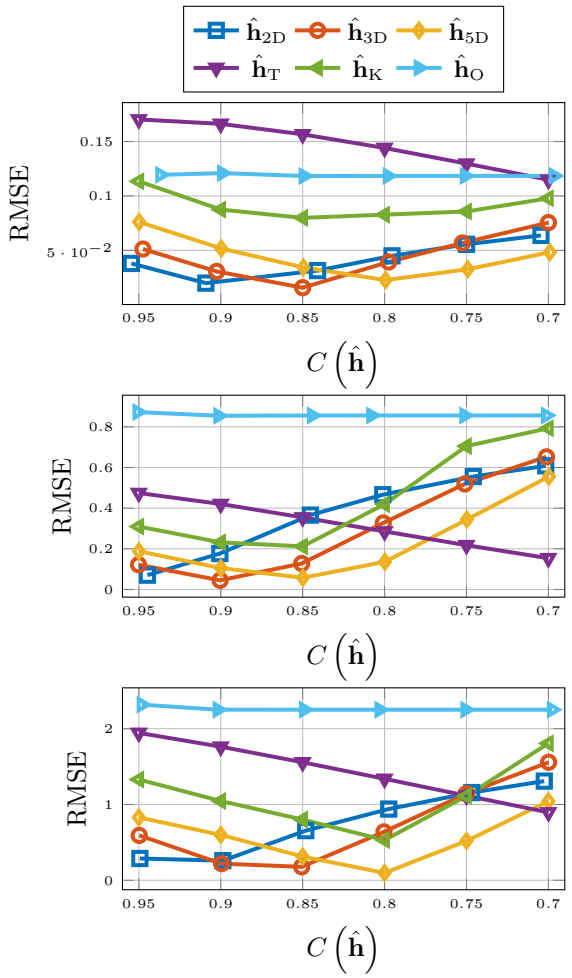

Figure 4: RMSE for T_{60} for SMARD (top), AIL (middle), and SAL (bottom)

where g is the considered measure, and n_{RIR} denotes the number of RIRs used in the evaluation. We alert the reader that we in Section 5 will consider RMSE in linear scale for certain measures and in logarithmic scale for other measures, depending on what best highlights the difference in performance between the considered compression methods. All considered quantities except echo density are scalar, making the computation of the RMSE straightforward. Echo density, however, is a discrete-time sequence. There the RMSE will be computed as

$$\text{RMSE}_{\text{ED}}\left(\hat{\mathbf{h}}\right) = \sqrt{\sum_{j=1}^{n_{\text{RIR}}} \|\mathbf{h}_{\text{ED}}^{j} - \hat{\mathbf{h}}_{\text{ED}}^{j}\|_2^2 \Big/ n_{\text{RIR}} n_{\text{ED}}}, \tag{30}$$

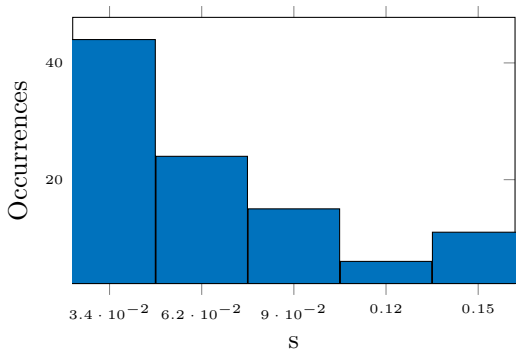

Figure 5: Histogram of differences in T_{60} estimate, $T_{60}(\hat{\mathbf{h}}_{3D}) - T_{60}(\mathbf{h})$, for 3-D tensor approximation and recorded RIR.

where $\mathbf{h}_{\text{ED}}^{j} = \left[\eta(1), \quad \eta(2), \quad \ldots, \quad \eta(n_{\text{ED}})\right]^T$ denotes the echo density profile of the jth RIR, and n_{ED} the length of the echo density profile.

4.2 Signal-based Objective Measures

Next, we present measures of output signal degradation. The ultimate goal of any acoustic signal enhancement or reproduction task is to achieve good signal quality. One way to measure this is by using subjective listening test. These tests are, however, expensive, tedious, and time consuming [47,60]. Therefore, several objective measures have been developed to predict the outcome of subjective listening tests. The frequency-weighted log-spectral signal distortion (SD) [61] is a perceptually weighted objective measure of distortion of a sound signal, w.r.t. a reference signal

$$\text{SD}(t) = \sqrt{\int_{f_l}^{f_u} w_{\text{ERB}}(f) \left(10 \log_{10} \frac{P_{\hat{\mathbf{y}}}(f,t)}{P_{\mathbf{y}}(f,t)}\right)^2 df}, \tag{31}$$

where $P_{\hat{\mathbf{y}}}$ and $P_{\mathbf{y}}$ are the short-term power spectra of $\hat{\mathbf{y}} = \mathbf{x} * \hat{\mathbf{h}}$ and $\mathbf{y} = \mathbf{x} * \mathbf{h}$, for a sound signal $\mathbf{x}$, respectively, and w_{ERB} is a frequency-weighting function, that gives equal weight to each auditory critical band between $f_l = 3000$ Hz and $f_u = 6500$ Hz. In Section 5, we will present both the mean and maximum SD values for the respective scenarios.

Hines *et. al* introduced the Virtual Speech Quality Objective Listener (ViSQOL) [49,50], an objective measure for predicting the subjective assessment of perceived speech quality, based on the Neurogram Similarity Index Measure (NSIM) [62]. ViSQOL was subsequently extended to ViSQOLAudio [51], to comprise not only speech, but also audio and music signals, and has shown high correlation with the subjective listening test MUSHRA [63]. Narbutt *et al.* have extended

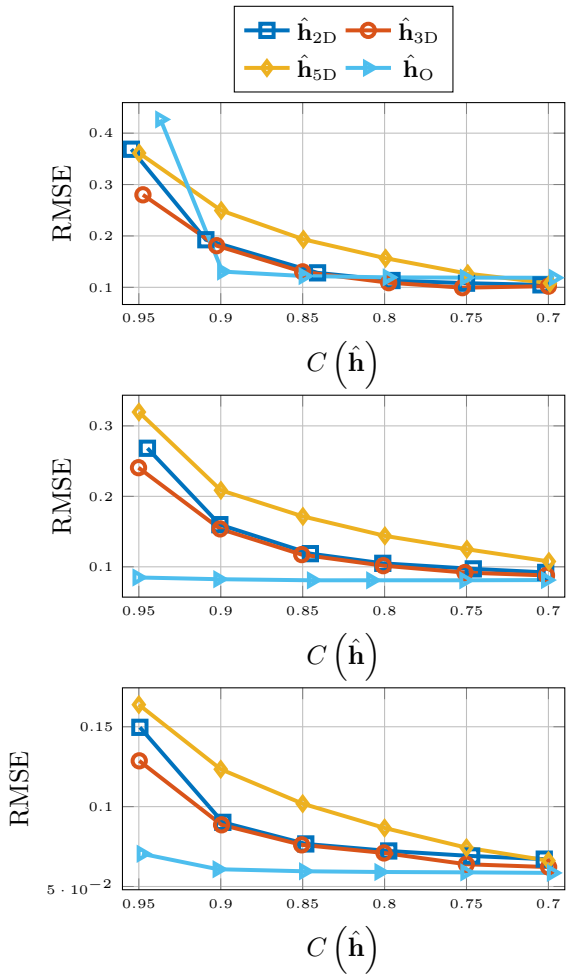

Figure 6: RMSE for echo density for SMARD (top), AIL (middle), and SAL (bottom)

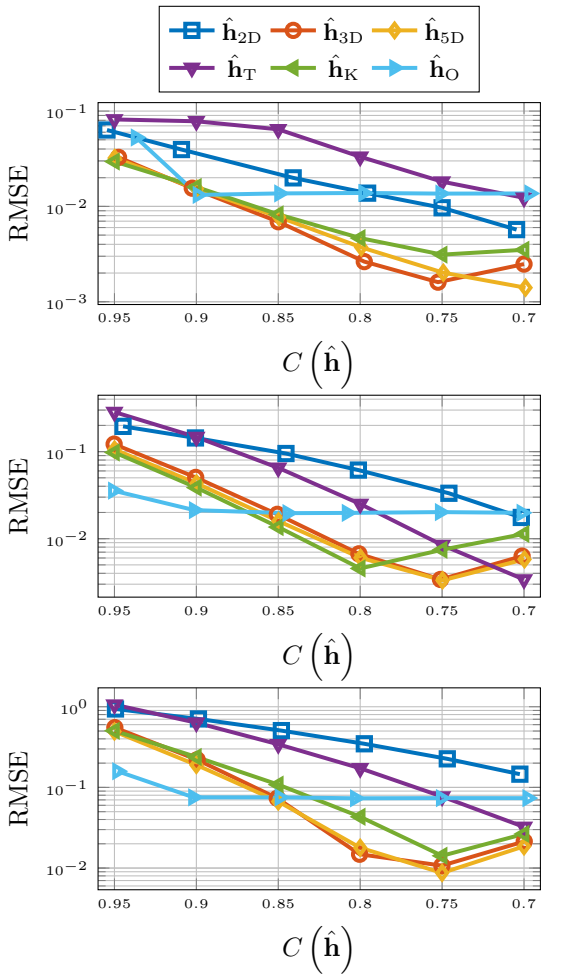

Figure 7: RMSE for EDT for SMARD (top), AIL (middle), and SAL (bottom)

ViSQOL and ViSQOLAudio to AMBIQUAL [64, 65], that aims to predict not only listening quality, but also localization accuracy, for spatial audio. We don't consider spatial audio in this work, and will therefore not use AMBIQUAL. In addition to the aforementioned acoustic qualities and measures, there are several other measures concerning perceived speech quality, such as PESQ [66] and POLQA [67]. These are intended to predict the perceived quality of speech, rather than audio or music, and will not be considered here.

5 Numerical Results

To compare the performance of the here investigated methods, we apply them to three different datasets of RIRs, with varying reverberation time. First we apply it to the single- and multichannel audio recordings database (SMARD) [48], which contains RIRs from a listening room with a reverberation time of approximately 0.15 s, sampled at 48 kHz. Next, we apply the methods to the two different datasets from the MYRiAD database [68]. The first one is from the Alamire Interactive Laboratory (AIL), which has a reverberation time of 0.5 s, and the second one is from the SONORA Audio Laboratory (SAL), with a reverberation time of 2.1 s. These are sampled at 44.1 kHz.

For the low-rank methods, the matricization or tensorization of the RIRs brings about the question of the size of the dimensions. For a D-dimensional tensorization, it is required that $\prod_{d=1}^{D} n_{s_d} = n_h$, but this can be achieved in several different ways. The impact of the size of the dimensions is beyond the scope of this paper, and we will here present only square matricizations and tensorizations, i.e., $n_{s_1} = n_{s_2} = \cdots = n_{s_D}$. As a consequence of this, we must have that $n_{s_d} = \sqrt[D]{n_h} \in \mathbb{N}$. For this reason, the length of the RIRs for the different compression meth-

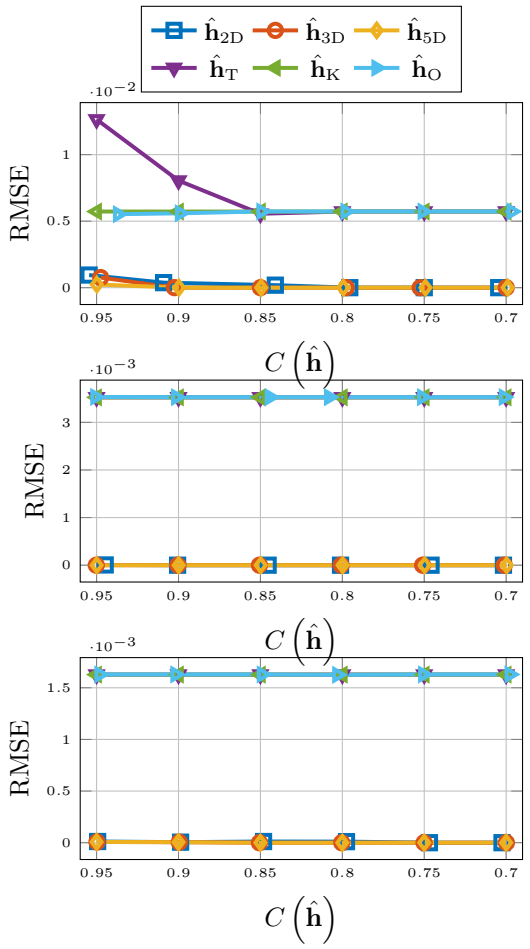

Figure 8: RMSE for TOA of direct component for SMARD (top), AIL (middle), and SAL (bottom)

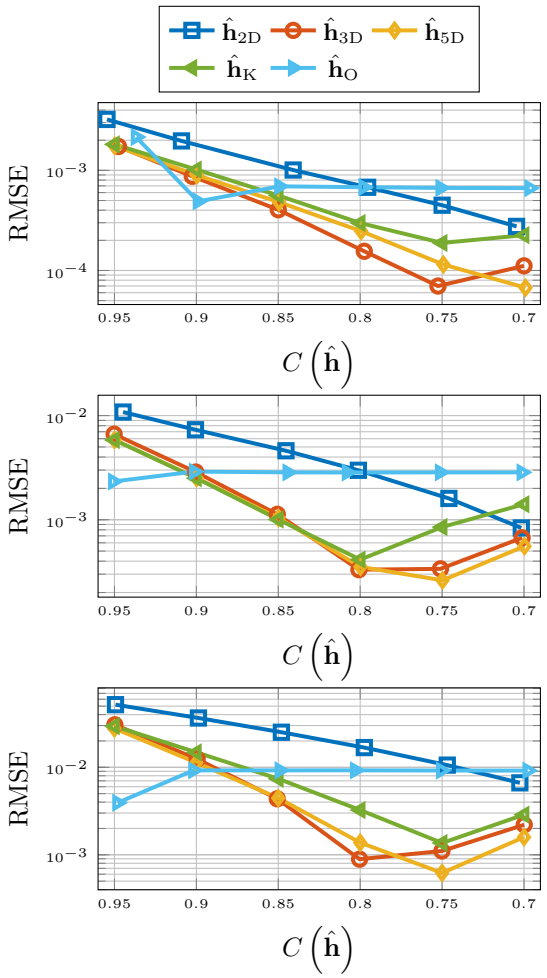

Figure 9: RMSE for center time for SMARD (top), AIL (middle), and SAL (bottom)

ods will vary slightly. We will here present the results for low-rank approximations of different dimensions, thresholding, truncation, and, as a benchmark, Opus. In order to be able to have RIR lengths in as close proximity as possible, we present low-rank approximations for $D = 2, 3,$ and 5, neglecting $D = 4$, as the length of the RIR for that dimension of tensorization would differ too much from the others. The RIR lengths used for the 2-D, 3-D, and 5-D approximations are denoted n_{h_2}, n_{h_3}, and n_{h_5}, respectively. The RIR length used for thresholding, truncation, and Opus is denoted n_h, and will be equal to the largest of n_{h_2}, n_{h_3}, and n_{h_5}, for the respective scenarios. The different RIR lengths used in the simulations are found in Table 2. We alert the reader that these lengths apply to both the approximation and their respective reference RIR, as some of the objective measures introduced in Section 4 require that the approximated RIR and the reference RIR are of equal length. For the generation of the output signals, the compressed RIRs are convolved with 5 different, randomly selected, 15 s snippets of music from EBU-SQAM [69]. When convolving these snippets of music with the RIRs from SMARD, the music was upsampled to 48 kHz using Matlab's *resample*, in order to have matching sampling frequencies.

We denote by $\Upsilon(\hat{\mathbf{h}})$ the number of coefficients needed to be stored for a certain compressed RIR $\hat{\mathbf{h}}$, and remind the reader that for the low-rank approximations, $\Upsilon(\hat{\mathbf{h}}) = R\sum_{d=1}^{D} n_{s_d}$. For all the compression methods except Opus, the number of coefficients stored coincides with the number of multiply-add instructions needed to carry out time-domain convolution with the approximated RIR. For the original RIR, this number is n_h. Therefore, by

$$C(\hat{\mathbf{h}}) = 1 - \frac{\Upsilon(\hat{\mathbf{h}})}{n_h}, \tag{32}$$

Table 2: RIR lengths used for the different data sets

Name	n_{RIR}	n_{h_2}	n_{h_3}	n_{h_5}
SMARD	100	$88^2 = 7744$	$20^3 = 8000$	$6^5 = 7776$
AIL	40	$181^2 = 32761$	$32^3 = 32768$	$8^5 = 32768$
SAL	20	$316^2 = 99856$	$47^3 = 103823$	$10^5 = 100000$

where $C(\hat{\mathbf{h}}) \in [0, 1)$, we denote both *compression rate* and *complexity reduction*. For $C(\hat{\mathbf{h}}) = 0$ there is no compression or complexity reduction, whereas for $C(\hat{\mathbf{h}})$ closer to 1, the degree of complexity reduction is larger. We provide simulations in the range from $C(\hat{\mathbf{h}}) = 0.7$ to $C(\hat{\mathbf{h}}) = 0.95$, as these are the minimum and maximum values of compression supported by Opus, for all the sets of RIRs considered here, when using Matlab's built-in function *audiowrite*.

RIRs should ideally be estimated from noiseless measurements, but this condition is often not met in practice [70–72]. As the RIRs used in this paper are taken from databases of real-life recorded RIRs, they will contain some measurement noise. However, to simulate a realistic environment, white Gaussian noise was added to each recorded and truncated RIR before compression and convolution. The power of the noise was adjusted to yield a signal-to-noise ratio (SNR) of 20 dB, where

$$\text{SNR}_{\text{dB}} = 10 \log_{10} \left(\frac{P_R}{P_N} \right), \tag{33}$$

where P_R and P_N denote the power of the RIR without the noise, and the power of the noise, respectively. The ground-truth values of the quantities considered in this section is computed with respect to truncated RIR, before the noise is added.

On a couple of occasions, the performance of one, or several, compression methods was significantly worse than the other methods. In those cases, these approximations have been left out of the figures, as including them would significantly impact the scaling of the figure, and prevent the reader from noticing the differences between the more competitive methods. When this has been done, remarks have been made in the corresponding subsection to alert the reader.

5.1 Normalized Misalignment

As can be seen in Fig. 3, in terms of normalized misalignment for the RIR compression, truncation and 2-D matricization falls short. However, 3-D tensorization, 5-D tensorization, and thresholding are all outperforming Opus.

5.2 Reverberation Time T_{60}

Compression based on low-rank approximation or thresholding also performs very well when it comes to the preservation of the reverberation time T_{60}. This is displayed in Fig. 4, where we observe a consistent outperformance of Opus. The unexpected performance deterioration for the low-rank approximation and thresholding is due to the added noise. Overestimation of T_{60} for noisy RIRs is a well-known phenomenon [73, 74]. This is due to a slower drop-off of the decay curve (22). The approximations serve as denoising but for lower values of compression there is still a systematic overestimation of the reverberation time. This is illustrated in Fig. 5, where histogram of the differences between the T_{60} estimates for the 3-D tensor approximation and that of the measured RIR, for the RIRs of SMARD, at the compression rate of 0.7, is displayed. We alert the reader that these are differences and not absolute differences, i.e. the fact that all numbers are positive shows the consistent overestimation. Preliminary simulations showed that this systematic overestimation could partly be alleviated by estimating the T_{60} a shorter time interval, i.e., corresponding to the decay from -5 dB to -25 dB, but not entirely.

5.3 Echo Density

When it comes to preserving echo density, displayed in Fig. 6, Opus is the best of the compared compression methods for longer RIRs. For short RIRs, 2-D matrix approximation and 3-D tensor approximation outperforms Opus, but 5-D tensor approximation does not. Truncation and thresholding are not included in Fig. 6 due to poor performance.

5.4 Early Decay Time

The performance of the different compression methods with respect to preserving EDT is shown in Fig. 7. For this measure, truncation and 2-D matricization performs worst for all considered cases. Opus works better for longer RIRs and for higher compression rates, but for shorter RIRs, and all but the highest compression rates, thresholding, and 3-D and 5-D tensorization are better options.

5.5 TOA of Direct Component

For the preservation of the TOA of the direct component, there is a clear discrepancy between the compression methods based on low-rank approximation and the other methods. This is evident from Fig. 8, where the results are displayed.

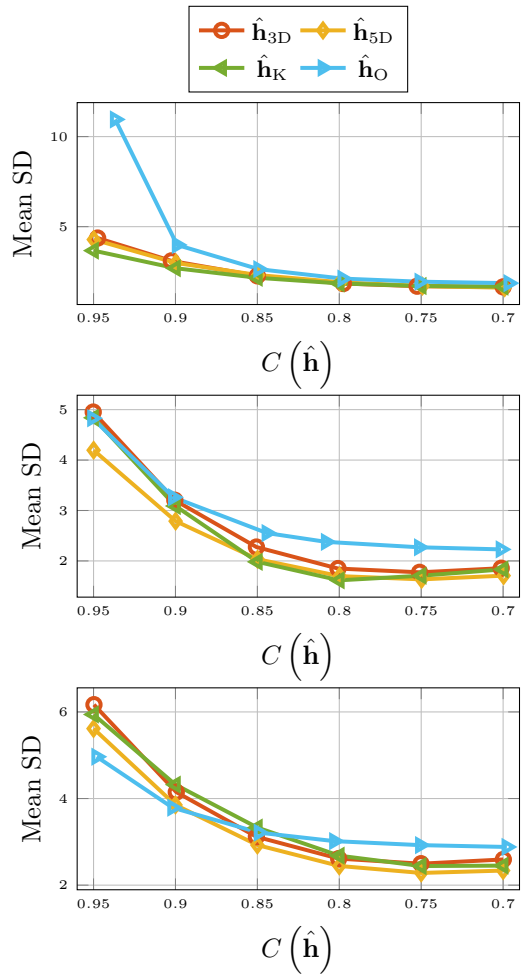

Figure 10: Mean signal distortion for SMARD (top), AIL (middle), and SAL (bottom)

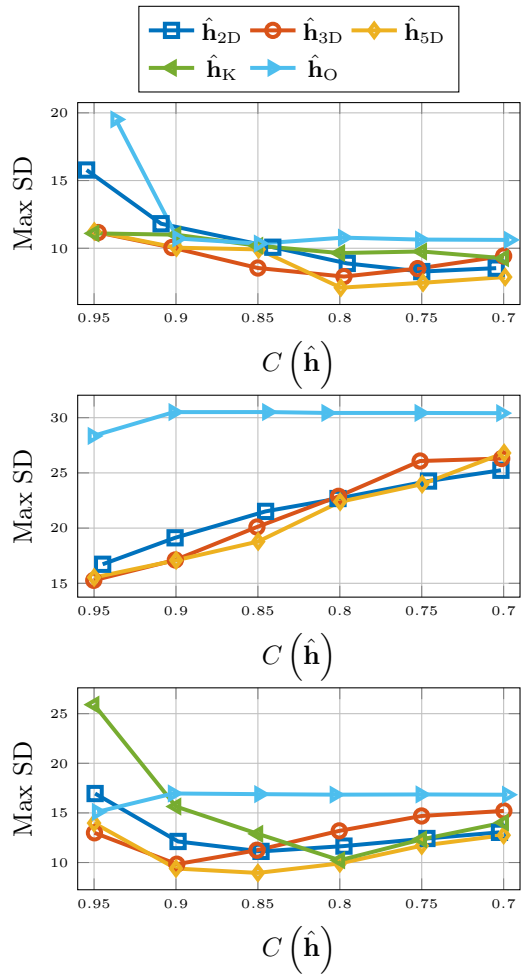

Figure 11: Maximum signal distortion for SMARD (top), AIL (middle), and SAL (bottom)

5.6 Center Time

In Fig. 9 we see the RMSE for the center time. There it can be observed that the 2-D matrix approximation does not perform on the level of Opus, but thresholding, and the higher-order tensor approximations do, for all but the highest compression rates. The performance of compression by truncation has been left out of the figure.

5.7 Signal Distortion

The results for the mean SD are better for the higher-order low-rank methods and thresholding, compared to Opus, except for the highest compression rates for the longest RIRs. This can be seen in Fig. 10. The results for 2-D matricization and truncation was yet again worse and left out of the plot to better show the difference between the other compression methods.

As for the maximum SD, displayed in Fig. 11, truncation had to be left out of the plots and for the AIL dataset, thresholding had to be left out too, due to their poor performance for higher compression rates. The higher-order low-rank approximation perform better than Opus for all the considered values of the compression rate.

5.8 ViSQOLAudio

In Fig. 12 the ViSQOLAudio scores for varying compression rate are displayed. It is only for high compression rates of very long RIRs where Opus is a better option than 3-D tensorization, 5-D tensorization, and thresholding. For ViSQOLAudio, the results for 2-D matricization and truncation were left out of Fig. 12 due to poor performance.

6 Conclusions

In this work we have considered different RIR approximation methods for the purpose of RIR compres-

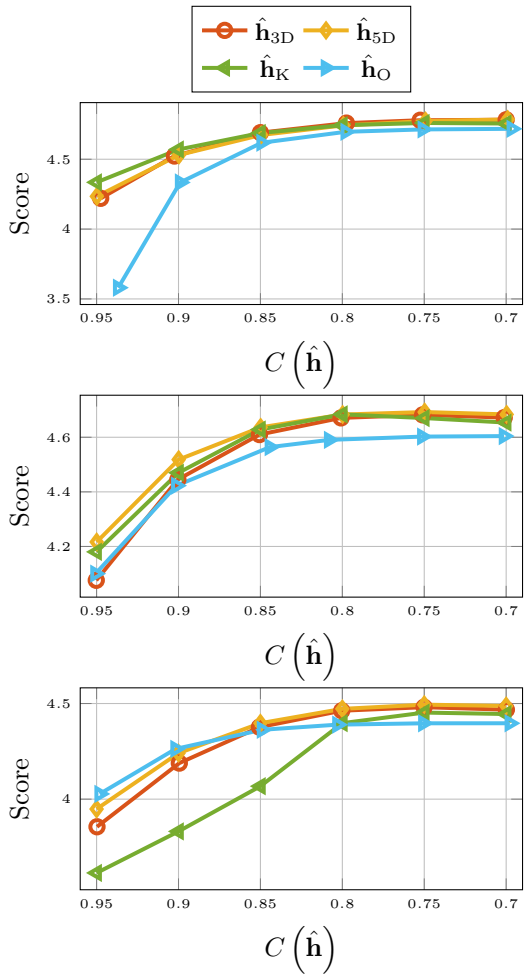

Figure 12: ViSQOLAudio for SMARD (top), AIL (middle), and SAL (bottom)

sion, aiming to save data storage and accelerate time-domain convolution. It was found that RIR truncation performs worst in almost all scenarios considered and can therefore not be recommended. With the exception of echo density, the RIR compression by thresholding generally preserves well the RIR qualities considered here, compared to the state-of-the-art Opus. For the low-rank approximation methods, 2-D matricization falls short on certain measures, such as mean signal distortion, and ViSQOLAudio. The 3-D and 5-D tensor approximations generally outperforms thresholding and they are more robust, as there was no considered scenario or measure where they preformed significantly worse than the other methods, and they perform better than thresholding with respect to the signal-based measures. Much like thresholding, 3-D and 5-D tensor approximations can't compete with Opus when it comes to preserving echo density, and for the highest level of compression rate, Opus is also better when it comes to preserving EDT and center time. For all other considered measures and scenarios, 3-D and 5-D tensor approximations are as good, or better, than Opus. Add to this the fact that the low-rank tensor approximations are amenable to fast time-domain convolution, and they stand out as the superior choice compared to Opus.

Future research should mainly focus on three open questions. Firstly, investigating whether the promising results for the objective measures considered here will translate into superior performance also in subjective listening tests. Secondly, the fact that the low-rank approximations preserve the TOA of the direct component almost flawlessly indicates that these approximations could be very useful also in the context of spatial RIRs, which needs to be further explored. Finally, the occasional discrepancy in performance between the 3-D and 5-D tensorization methods is not yet well enough understood, and needs to be further investigated.

Acknowledgements
Not applicable

Funding
This research work was carried out at the ESAT Laboratory of KU Leuven. The research leading to these results has received funding from the European Research Council under the European Union's Horizon 2020 research and innovation program / ERC Consolidator Grant: SONORA (no. 773268). This paper reflects only the authors' views and the Union is not liable for any use that may be made of the contained information.

Abbreviations
Not applicable

Availability of data and materials
Not applicable

Ethics approval and consent to participate
Not applicable

Competing interests
The authors declare that they have no competing interests.

Consent for publication
Not applicable

Authors' contributions
All authors jointly developed the methodology and designed the computer simulations presented. MJ implemented the algorithms and computer simulations, and all authors jointly interpreted the results obtained. MJ drafted the manuscript and all authors read and reviewed the final manuscript.

Authors' information
Not applicable

Author details
[1]Department of Electrical Engineering (ESAT/STADIUS), KU Leuven, Leuven, Belgium. [2]Department of Information and Communications Engineering, Aalto University, Espoo, Finland.

References

1. Evers, C., Löllmann, H.W., Mellmann, H., Schmidt, A., Barfuss, H., Naylor, P.A., Kellermann, W.: The LOCATA challenge: Acoustic source localization and tracking. IEEE/ACM Trans. Audio Speech Lang. Process. **28**, 1620–1643 (2020). doi:10.1109/TASLP.2020.2990485
2. Naylor, P.A., Gaubitch, N.D.: Speech Dereverberation. Springer, London, United Kingdom (2010)
3. Vorländer, M.: Auralization: Fundamentals of Acoustics, Modelling, Simulation, Algorithms and Acoustic Virtual Reality. Springer, Switzerland (2020)
4. Brinkmann, F., Aspöck, L., Ackermann, D., Lepa, S., Vorländer, M., Weinzierl, S.: A round robin on room acoustical simulation and auralization. The Journal of the Acoustical Society of America **145**(4), 2746–2760 (2019)
5. Gannot, S., Vincent, E., Markovich-Golan, S., Ozerov, A.: A consolidated perspective on multimicrophone speech enhancement and source separation. IEEE/ACM Trans. Audio Speech Lang. Process. **25**(4), 692–730 (2017). doi:10.1109/TASLP.2016.2647702
6. Goetze, S., Albertin, E., Kallinger, M., Mertins, A., Kammeyer, K.-D.: Quality assessment for listening-room compensation algorithms. In: 2010 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 2450–2453 (2010). doi:10.1109/ICASSP.2010.5496301
7. Elko, G.W., Diethorn, E., Gaensler, T.: Room impulse response variation due to thermal fluctuation and its impact on acoustic echo cancellation. In: Proc. 2003 Int. Workshop Acoustic Echo Noise Control (IWAENC '03), Kyoto, Japan (2003)
8. Mourjopoulos, J., Paraskevas, M.A.: Pole and zero modeling of room transfer functions. Journal of Sound and Vibration **146**(2), 281–302 (1991)
9. Vairetti, G., De Sena, E., Catrysse, M., Jensen, S.H., Moonen, M., van Waterschoot, T.: A scalable algorithm for physically motivated and sparse approximation of room impulse responses with orthonormal basis functions. IEEE/ACM Trans. Audio Speech Lang. Process. **25**(7), 1547–1561 (2017)
10. Huszty, C., Bukuli, N., Torma, Á., Augusztinovicz, F.: Effects of filtering of room impulse responses on room acoustics parameters by using different filter structures. J. Acoust. Soc. Amer. **123**, 3617 (2008)
11. Vairetti, G.: Efficient parametric modeling, identification and equalization of room acoustics. PhD thesis, KU Leuven (2018)
12. Ngia, L.S.H.: Recursive identification of acoustic echo systems using orthonormal basis functions. IEEE Trans. Speech Audio Process. **11**(3), 278–293 (2003)
13. Rossing, T.: Springer Handbook of Acoustics. Springer, New York, NY, USA (2014)
14. Shi, K., Ma, X., Tong Zhou, G.: An efficient acoustic echo cancellation design for systems with long room impulses and nonlinear loudspeakers. Signal Processing **89**(2), 121–132 (2009)
15. Krishnan, L., Teal, P.D., Betlehem, T.: A robust sparse approach to acoustic impulse response shaping. In: 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 738–742 (2015). doi:10.1109/ICASSP.2015.7178067
16. Hacihabiboglu, H., De Sena, E., Cvetkovic, Z., Johnston, J., Smith III, J.O.: Perceptual spatial audio recording, simulation, and rendering: An overview of spatial-audio techniques based on psychoacoustics. IEEE Signal Processing Magazine **34**(3), 36–54 (2017). doi:10.1109/MSP.2017.2666081
17. Katz, B.F.G., Murphy, D., Farina, A.: The past has ears (PHE): XR explorations of acoustic spaces as cultural heritage. In: De Paolis, L.T., Bourdot, P. (eds.) Augmented Reality, Virtual Reality, and Computer Graphics, pp. 91–98. Springer, Cham (2020)
18. Ajdler, T., Sbaiz, L., Vetterli, M.: The plenacoustic function and its sampling. IEEE Trans. Signal Process. **54**(10), 3790–3804 (2006). doi:10.1109/TSP.2006.879280
19. Rafaely, B., Tourbabin, V., Habets, E., Ben-Hur, Z., Lee, H., Gamper, H., Arbel, L., Birnie, L., Abhayapala, T., Samarasinghe, P.: Spatial audio signal processing for binaural reproduction of recorded acoustic scenes - review and challenges. Acta Acust. **6**, 47 (2022)
20. Gupta, R., He, J., Ranjan, R., Gan, W.-S., Klein, F., Schneiderwind, C., Neidhardt, A., Brandenburg, K., Välimäki, V.: Augmented/mixed reality audio for hearables: Sensing, control, and rendering. IEEE Signal Processing Magazine **39**(3), 63–89 (2022). doi:10.1109/MSP.2021.3110108
21. Schissler, C., Stirling, P., Mehra, R.: Efficient construction of the spatial room impulse response. In: 2017 IEEE Virtual Reality (VR), pp. 122–130 (2017). doi:10.1109/VR.2017.7892239
22. Møller, M.B., Østergaard, J.: A moving horizon framework for sound zones. IEEE/ACM Trans. Audio Speech Lang. Process. **28**, 256–265 (2020). doi:10.1109/TASLP.2019.2951995
23. Brunnström, J., Jälmby, M., van Waterschoot, T., Moonen, M.: Fast low-rank filtered-x least mean squares for multichannel active noise control. https://ftp.esat.kuleuven.be/pub/stadius/jbrunnst/2023_asilomar_low_rank_fxlms_006.pdf (2023)
24. Carôt, A., Werner, C.: Network music performance-problems, approaches and perspectives. In: Proceedings of the "Music in the Global Village"-Conference, Budapest, Hungary, vol. 162, pp. 10–23 (2007)
25. Välimaki, V., Parker, J.D., Savioja, L., Smith, J.O., Abel, J.S.: Fifty years of artificial reverberation. IEEE/ACM Trans. Audio Speech Lang. Process. **20**(5), 1421–1448 (2012). doi:10.1109/TASL.2012.2189567
26. Cooley, J.W., Tukey, J.W.: An algorithm for the machine calculation of complex fourier series. Mathematics of Computation **19**, 297–301 (1965)
27. Stockham, T.G.: High-speed convolution and correlation. In: Proceedings of the April 26-28, 1966, Spring Joint Computer Conference. AFIPS '66 (Spring), pp. 229–233. Association for Computing Machinery, New York, NY, USA (1966). doi:10.1145/1464182.1464209
28. Wefers, F.: Partitioned Convolution Algorithms for Real-Time Auralization. Logos Verlag, DEU (2015)
29. Primavera, A., Cecchi, S., Romoli, L., Peretti, P., Piazza, F.: A low latency implementation of a non-uniform partitioned convolution algorithm for room acoustic simulation. Signal, Image and Video Processing **8**(5), 985–994 (2014)
30. Vorländer, M., Schröder, D., Pelzer, S., Wefers, F.: Virtual reality for architectural acoustics. Journal of Building Performance Simulation **8**(1), 15–25 (2015)
31. Lee, W.-C., Liu, C.-M., Yang, C.-H., Guo, J.-I.: Fast perceptual convolution for room reverberation. In: 6th International Conference on Digital Audio Effects (DAFx-03), London, United Kingdom (2003)
32. Jillings, N., Reiss, J.D., Stables, R.: Zero-delay large signal convolution using multiple processor architectures. In: Proc. 2017 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA), pp. 339–343 (2017)
33. Holm-Rasmussen, B., Lehtonen, H., Välimäki, V.: A new reverberator based on variable sparsity convolution. In: Proc. 16th Int. Conf. Digital Audio Effects (DAFx-13), Maynooth, Ireland (2013)
34. Jälmby, M., Elvander, F., van Waterschoot, T.: Low-rank tensor modeling of room impulse responses. In: 2021 29th European Signal Processing Conference (EUSIPCO), pp. 111–115 (2021). doi:10.23919/EUSIPCO54536.2021.9616075
35. Jälmby, M., Elvander, F., van Waterschoot, T.: Low-rank room impulse response estimation. IEEE/ACM Trans. Audio Speech Lang. Process. **31**, 957–969 (2023). doi:10.1109/TASLP.2023.3240650
36. Jälmby, M., Elvander, F., van Waterschoot, T.: Multi-channel Low-rank Convolution of Jointly Compressed Room Impulse Responses. https://ftp.esat.kuleuven.be/pub/stadius/mjalmby/23-150.pdf (2023)
37. Atkins, J., Strauss, A., Zhang, C.: Approximate convolution using partitioned truncated singular value decomposition filtering. In: Proc. 2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 176–180 (2013). doi:10.1109/ICASSP.2013.6637632
38. Jälmby, M., Elvander, F., van Waterschoot, T.: Fast low-latency convolution by low-rank tensor approximation. In: 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 1–5 (2023). doi:10.1109/ICASSP49357.2023.10095908
39. Jaderberg, M., Vedaldi, A., Zisserman, A.: Speeding up Convolutional Neural Networks with Low Rank Expansions. arXiv: 1405.3866 (2014)
40. Sorber, L., Van Barel, M., De Lathauwer, L.: Optimization-based algorithms for tensor decompositions: Canonical polyadic

decomposition, decomposition in rank-$(L_r, L_r, 1)$ terms, and a new generalization. SIAM Journal on Optimization **23**(2), 695–720 (2013). doi:10.1137/120868323. https://doi.org/10.1137/120868323
41. Vervliet, N., Debals, O., Sorber, L., Van Barel, M., De Lathauwer, L.: Tensorlab 3.0. Available online. https://www.tensorlab.net (2016)
42. Valin, J.-M., Vos, K., Terriberry, T.: Definition of the Opus audio codec. https://www.rfc-editor.org/rfc/rfc6716 (2012)
43. Valin, J.-M., Maxwell, G., Terriberry, T.B., Vos, K.: High-quality, low-delay music coding in the Opus codec. In: Proc. 135th AES Convention (2013)
44. Vos, K., Jensen, S., Soerensen, K.: SILK Speech Codec, https://datatracker.ietf.org/doc/html/draft-vos-silk-02 (2010)
45. Valin, J.-M.-M., Terriberry, T.B., Maxwell, G.: A full-bandwidth audio codec with low complexity and very low delay. In: 2009 17th European Signal Processing Conference (EUSIPCO), pp. 1254–1258 (2009)
46. Valin, J.-M., Terriberry, T.B., Montgomery, C., Maxwell, G.: A high-quality speech and audio codec with less than 10-ms delay. IEEE Trans. Audio Speech Lang. Process. **18**(1), 58–67 (2010). doi:10.1109/TASL.2009.2023186
47. Ren, H., Ritz, C., Zhao, J., Jang, D.: Impact of compression on the performance of the room impulse response interpolation approach to spatial audio synthesis. In: 2022 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), pp. 442–448 (2022). doi:10.23919/APSIPAASC55919.2022.9980324
48. Nielsen, J.K., Jensen, J.R., Jensen, S.H., Christensen, M.G.: The single- and multichannel audio recordings database (SMARD). In: Proc. 2014 Int. Workshop Acoustic Signal Enhancement (IWAENC '14), Antibes, France (2014)
49. Hines, A., Skoglund, J., Kokaram, A., Harte, N.: ViSQOL: The virtual speech quality objective listener. In: IWAENC 2012; International Workshop on Acoustic Signal Enhancement, pp. 1–4 (2012)
50. Hines, A., Skoglund, J., Kokaram, A.C., Harte, N.: Visqol: an objective speech quality model. EURASIP Journal on Audio, Speech, and Music Processing **2015**(1), 13 (2015)
51. Hines, A., Gillen, E., Kelly, D., Skoglund, J., Kokaram, A., Harte, N.: ViSQOLaudio: An objective audio quality metric for low bitrate codecs. The Journal of the Acoustical Society of America **137 (6)**, 449–455 (2015)
52. Schroeder, M.R.: New method of measuring reverberation time. J. Acoust. Soc. Am. **37**(3), 409–412 (1965)
53. Rumsey, F.: Spatial Audio. Focal Press, Oxford, United Kingdom (2001)
54. Abel, J.S., Huang, P.: A simple, robust measure of reverberation echo density. In: Audio Engineering Society Convention 121 (2006). Audio Engineering Society
55. De Sena, E., Hacihabiboglu, H., Cvetkovic, Z., Smith, J.O.: Efficient synthesis of room acoustics via scattering delay networks. IEEE/ACM Trans. Audio Speech Lang. Process. **23**(9), 1478–1492 (2015). doi:10.1109/TASLP.2015.2438547
56. Karjalainen, M., Järvelainen, H.: More about this reverberation science: Perceptually good late reverberation. In: Proceedings of the 111th Audio Engineering Society Convention, New York, NY (2011)
57. Kuttruff, H.: Room Acoustics. Spon Press, London (2009)
58. MacWilliam, K., Elvander, F., van Waterschoot, T.: Simultaneous acoustic echo sorting and 3-d room geometry inference. In: ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 1–5 (2023). doi:10.1109/ICASSP49357.2023.10096005
59. Rosseel, H., van Waterschoot, T.: Improved acoustic source localization by time delay estimation with subsample accuracy. In: 2021 Immersive and 3D Audio: from Architecture to Automotive (I3DA), pp. 1–8 (2021). doi:10.1109/I3DA48870.2021.9610902
60. Cartwright, M., Pardo, B., Mysore, G.J., Hoffman, M.: Fast and easy crowdsourced perceptual audio evaluation. In: 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 619–623 (2016). doi:10.1109/ICASSP.2016.7471749
61. Spriet, A., Eneman, K., Moonen, M., Wouters, J.: Objective measures for real-time evaluation of adaptive feedback cancellation algorithms in hearing aids. In: 2008 16th European Signal Processing Conference (EUSIPCO), pp. 1–5 (2008)
62. Hines, A., Harte, N.: Speech intelligibility prediction using a neurogram similarity index measure. Speech Communication **54**(2), 306–320 (2012). doi:10.1016/j.specom.2011.09.004
63. Rec.ITU-R.BS.1534-1:: Method for the Subjective Assessment of Intermediate Sound Quality (MUSHRA). International Telecommunication Union, Geneva (2003). International Telecommunication Union
64. Narbutt, M., Allen, A., Skoglund, J., Chinen, M., Hines, A.: Ambiqual - a full reference objective quality metric for ambisonic spatial audio. In: 2018 Tenth International Conference on Quality of Multimedia Experience (QoMEX), pp. 1–6 (2018). doi:10.1109/QoMEX.2018.8463408
65. Narbutt, M., Skoglund, J., Allen, A., Chinen, M., Barry, D., Hines, A.: Ambiqual: Towards a quality metric for headphone rendered compressed ambisonic spatial audio. Applied Sciences **10**(9) (2020). doi:10.3390/app10093188
66. Rix, A.W., Beerends, J.G., Hollier, M.P., Hekstra, A.P.: Perceptual evaluation of speech quality (PESQ) - a new method for speech quality assessment of telephone networks and codecs. In: 2001 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), vol. 2, pp. 749–7522 (2001). doi:10.1109/ICASSP.2001.941023
67. Beerends, J., Schmidmer, C., Berger, J., Obermann, M., Ullmann, R., Pomy, J., Keyhl, M.: Perceptual objective listening quality assessment (POLQA), the third generation ITU-T standard for end-to-end speech quality measurement part I — temporal alignment. Journal of the Audio Engineering Society **61**(6), 366–384 (2013)
68. Dietzen, T., Ali, R., Taseska, M., van Waterschoot, T.: MYRiAD: a multi-array room acoustic database. EURASIP Journal on Audio, Speech, and Music Processing **2023**(1), 17 (2023). doi:10.1186/s13636-023-00284-9
69. Waters, G.: Sound quality assessment material—recordings for subjective tests: User's handbook for the EBU–SQAM compact disk. European Broadcasting Union (EBU), Tech. Rep, 1–13 (1988)
70. Paulo, J.P., Martins, C.R., Bento Coelho, J.L.: A hybrid MLS technique for room impulse response estimation. Applied Acoustics **70**(4), 556–562 (2009)
71. Ćirić, D.G., Janković, M.: Correction of room impulse response truncation based on a nonlinear decay model. Applied Acoustics **132**, 210–222 (2018)
72. Crocco, M., Del Bue, A.: Room impulse response estimation by iterative weighted L1-norm. In: 2015 23rd European Signal Processing Conference (EUSIPCO), pp. 1895–1899 (2015). doi:10.1109/EUSIPCO.2015.7362713
73. Chen, M., Lee, C.-M.: The optimal determination of the truncation time of non-exponential sound decays. Buildings **12**(5) (2022). doi:10.3390/buildings12050697
74. Gaubitch, N.D., Loellmann, H.W., Jeub, M., Falk, T.H., Naylor, P.A., Vary, P., Brookes, M.: Performance comparison of algorithms for blind reverberation time estimation from speech. In: IWAENC 2012; International Workshop on Acoustic Signal Enhancement, pp. 1–4 (2012)