Received XX Month, XXXX; revised XX Month, XXXX; accepted XX Month, XXXX; Date of publication XX Month, XXXX; date of current version XX Month, XXXX.

Digital Object Identifier 10.1109/XXXX.2022.1234567

Multi-channel Low-rank Convolution of Jointly Compressed Room Impulse Responses

Martin Jälmby[1], Student Member, IEEE, Filip Elvander[2], and Toon van Waterschoot [1], Member, IEEE

[1]STADIUS Center for Dynamical Systems, Department of Electrical Engineering, KU Leuven, Belgium
[2]Dept. of Information and Communications Engineering, Aalto University, Finland

Corresponding author: Martin Jälmby (email: martin.jalmby@esat.kuleuven.be).

This research work was carried out at the ESAT Laboratory of KU Leuven, in the frame of KU Leuven internal funds C14/21/075 and FWO projects G0A2721N and S005319N. 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.

ABSTRACT The room impulse response (RIR) describes the response of a room to an acoustic excitation signal and models the acoustic channel between a point source and receiver. RIRs are used in a wide range of applications, e.g., virtual reality. In such an application, the availability of closely spaced RIRs and the capability to achieve low latency are imperative to provide an immersive experience. However, representing a complete acoustic environment using a fine grid of RIRs is prohibitive from a storage point of view and without exploiting spatial proximity, acoustic rendering becomes computationally expensive. We therefore propose two methods for the joint compression of multiple RIRs, *Low-rank Compression of Room Impulse Responses for Low-latency Convolution* and *Low-rank Compression of Room Impulse Responses for Low-latency Convolution with Partially Invariant Transformation*, based on the generalized low-rank approximation of matrices (GLRAM), for the purpose of efficiently storing RIRs and allowing for low-latency convolution, for which we propose an algorithm. We show how one of the components of the GLRAM decomposition is virtually invariant to the change of position of the source throughout the room and how this can be exploited in the modeling and convolution. In simulations we show how this offers high compression, with less quality degradation than for comparable benchmark methods.

INDEX TERMS Convolution, low-rank modeling, room impulse responses

I. INTRODUCTION

THE room impulse response (RIR) describes the impact of a room on an acoustic excitation signal played within the room and is used in a wide variety of applications [1]–[3]. Typically, the RIR is modeled either as a infinite impulse response (IIR) (see, e.g., [4], [5]), or as an finite impulse response (FIR) model (see, e.g., [4], [6]). The IIR model is generally more compact, but the filter parameters and be difficult to estimate, due to issues of instability [7], [8]. The FIR model is simple and straightforward, but the number of parameters needed is large, on the order of 10^3 for an office-sized room, and 10^4 or even 10^5 for a reverberant room such as concert hall, at a sampling rate of 48 kHz [9], [10]. This can be prohibitive from both a memory requirement and computational complexity point for view, when using the RIR for convolution [11]–[13]. The RIR is position-dependent, meaning that if the source or the receiver moves, the corresponding RIR changes. As a consequence, 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 [14]. Even for small rooms, the number of source/receiver configurations for which the RIR has to be stored will be in the millions, amounting to hundreds of gigabytes of data for a single room. Consequently, there is a need for compact representations of the RIRs of a room.

An application of RIRs, with a market that has seen a surge in recent years, and is expected to continue to grow, is that of simulated experiences, such as virtual reality (VR),

with purposes ranging from pure entertainment to decision making in building design, skill training, and therapy in mental health [15]–[18]. In VR it is desirable to allow the user to move around in the simulated space, thereby allowing for a more immersive experience. It is necessary to have closely spaced RIRs to be able to represent small sound sources in order for the experience to be immersive [16]. The storage of said RIRs can, however, be burdensome, due to the limited storage of real-world products [19]. Furthermore, the processing needs to be as light as possible, as most of the computational resources are used for the visual rendering [16]. This highlights the need for compact storage of, and fast low-latency convolution with, RIRs.

In order to provide alleviation in terms of storage and processing, we have in previous work considered low-rank models of matricizations and tensorizations of RIR vectors [20], how RIRs can be estimated on a low-rank form [21], how this low-rank structure can be leveraged in fast, low-latency time-domain convolution [22], how low-rank models preserve objective RIR qualities and perform with respect to objective signal-based measures [23], and for multi-channel active noise control [24]. In this paper we extend upon these ideas and propose the joint compression of multiple RIRs, for the purpose of saving storage space, as well as making them amenable to fast low-latency multi-channel convolution. This compression will be done using two different methods building upon the generalized low-rank approximation of matrices (GLRAM), introduced in [25]. We propose that the set of RIRs used to find the components of GLRAM does not need to be the same as the set of RIRs one aims to compress. This allows for scenarios where the compression is learnt on one set of RIRs and then later used on another set of RIRs, possibly unknown at the time when the compression was learnt. Huang *et al.* has in [26] considered system identification from input-output data, of RIRs corresponding to adjacent source positions, on the form of a tensor decomposition.

The contribution of this paper is fourfold. Firstly, we show how multiple RIRs of a room can be compressed jointly, with less quality degradation than comparable state-of-the-art methods, using joint low-rank representations. Secondly, we show how the components of this compression vary throughout the room, and how this insight can be leveraged in the modeling. Thirdly, we demonstrate how the compression can be learnt using a set of RIRs, and then used to compress a different set of RIRs. Finally, we propose an algorithm for multi-channel low-rank convolution with the jointly compressed RIRs, without the need to decompress these.

This paper is organized as follows: In Section II the signal model and the proposed algorithms are presented. Numerical results are presented in Section III. Finally, in Section IV, conclusions are presented and possible areas for future research are pointed out.

A. NOTATION

We denote scalars, vectors, and matrices, by lowercase (e.g., h), bold lowercase (e.g., $\mathbf{h}$), and bold uppercase letters (e.g., $\mathbf{H}$), respectively. Sets are denoted by calligraphic letters (e.g., $\mathcal{H}$), and the cardinality of a set is denoted $|\mathcal{H}|$. Linear operators are also denoted by uppercase calligraphic letters, but it will be obvious from context what is considered. $\mathbf{I}_n$ is an $n \times n$ identity matrix. The selection of one or several elements from a vector or matrix will be denoted by square brackets, e.g., $\mathbf{H}[m:p:n,j]$ is a vector containing every pth element from the mth till the nth row of the jth column of $\mathbf{H}$ (the omission of p indicates that every element between m and n is considered). The hat symbol, $\hat{\cdot}$, indicates an approximated quantity. 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. $\mathcal{I}(\mathbf{x}) = \begin{bmatrix} x(n_x), & x(n_x-1), & \ldots, & x(1) \end{bmatrix}^T$. Finally, eigenvalues and singular values are ordered in a non-increasing fashion, with respect to the magnitude.

II. SIGNAL MODEL, ALGORITHM, AND MOTIVATION

A. SIGNAL MODEL

We consider a discrete-time RIR $h(k)$, $k = 1, 2, \ldots, n_h$, arranged in the vector $\mathbf{h} \in \mathbb{R}^{n_h}$. The RIR can be modeled as a sum of decaying sinusoids (see [20] and references therein),

$$h(\mathbf{r}_r, \mathbf{r}_s, k) = \sum_{m=1}^{m_s} \mu_m(\mathbf{r}_r, \mathbf{r}_s) e^{-\beta_m n} \cos(\omega_m n + \phi_m), \quad (1)$$

for $k = 1, 2, \ldots n_h$. Here, μ_m denotes the initial amplitude, $\mathbf{r}_r, \mathbf{r}_s \in \mathbb{R}^3$ the position of the receiver and the source, respectively, $\beta_m \in \mathbb{R}_+$ the exponential decay constant, $\omega_m \in [0, \pi]$ the angular frequency, $\phi_m \in [0, 2\pi)$ the phase and $m_s \in \mathbb{N}$ is the number of decaying sinusoids used in the model. For the ease of notation, we will drop the dependence on $\mathbf{r}_r$ and $\mathbf{r}_s$ and refer to $h(\mathbf{r}_r, \mathbf{r}_s, k)$ as $h(k)$. Consider the matricization, or reshaping, of $\mathbf{h} = \begin{bmatrix} h(1) & h(2) & \ldots & h(n_h) \end{bmatrix}^T$ into a matrix $\mathbf{H} \in \mathbb{R}^{r \times c}$,

$$\mathbf{H} = \begin{bmatrix} h(1) & h(r+1) & \ldots & h(r(c-1)+1) \\ \vdots & \vdots & & \vdots \\ h(r) & h(2r) & \ldots & h(cr) \end{bmatrix}, \quad (2)$$

where it is assumed that $n_h = rc$. When a vector consisting of the sum of m_s discrete-time decaying sinusoids is reshaped into a matrix, that matrix will have rank $2m_s$ (see, e.g., [27]). The low-rank structure of the matricized RIR is something that can be exploited for purpose of compact storage [20], as well as fast, low-latency time-domain convolution [22]. We have in previous work focused on the low-rank structure of single RIRs. Here we look also to exploit the similarity of RIRs from closely spaced source or receiver positions, by considering joint low-rank approximation of multiple matricized RIRs. We will see that the original GLRAM decomposition is too restrictive for the purpose of joint compression of RIRs when they

correspond to source or receiver positions too far apart. However, by considering a modified form, good compression can be achieved and the approximated RIRs are amenable to fast low-latency time-domain convolution.

B. GLRAM

Consider a set of RIRs, which with (2) can be represented as a set of matrices $\mathbf{H}_j \in \mathbb{R}^{r\times c}$, $j = 1, 2, \ldots, n$. GLRAM constructs joint low-rank approximations $\mathbf{H}_j \approx \hat{\mathbf{H}}_j = \mathbf{L}\mathbf{D}_j\mathbf{R}^T, j = 1, 2, \ldots, n$, by minimizing the criterion

$$\begin{aligned} \underset{\mathbf{L},\mathbf{R},\mathbf{D}_j}{\text{minimize}} \quad & \sum_{j=1}^{n} \|\mathbf{H}_j - \mathbf{L}\mathbf{D}_j\mathbf{R}^T\|_F^2, \\ \text{s.t.} \quad & \mathbf{L}^T\mathbf{L} = \mathbf{I}_{\ell_1}, \mathbf{R}^T\mathbf{R} = \mathbf{I}_{\ell_2} \end{aligned} \tag{3}$$

where $\mathbf{L} \in \mathbb{R}^{r\times\ell_1}$, $\mathbf{R} \in \mathbb{R}^{c\times\ell_2}$, $\mathbf{D}_j \in \mathbb{R}^{\ell_1\times\ell_2}$. First $\mathbf{L}$ and $\mathbf{R}$ are found iteratively, and then the matrices $\mathbf{D}_j$ are found via $\mathbf{D}_j = \mathbf{L}^T\mathbf{H}_j\mathbf{R}$. The matrices $\mathbf{L}$ and $\mathbf{R}$ are common to all $\hat{\mathbf{H}}_j$, $j = 1, 2, \ldots, n$. While $\mathbf{L}$ and $\mathbf{R}$ are orthogonal, much like $\mathbf{U}$ and $\mathbf{V}$ of a traditional singular value decomposition (SVD), $\mathbf{H} = \mathbf{U}\mathbf{S}\mathbf{V}^T$, $\mathbf{D}_j$ is not necessarily, in contrast to $\mathbf{S}$, diagonal. It is also worth noting that ℓ_1 and ℓ_2 do not have to be equal, yielding extended modeling freedom. By considering $\ell_1 < r$, $\ell_2 < c$, and a large number of RIRs, n, the number of coefficients needed to represent the approximated matrices $\hat{\mathbf{H}}_j$, can be made significantly smaller than the number of coefficients needed to represent the original matrices $\mathbf{H}_j$. The original GLRAM algorithm can be found in [25].

C. PROPOSED ROOM COMPRESSION METHODS

We propose two distinct methods for the simultaneous compression of multi-channel RIRs, which will be introduced in this section. We will from hereon distinguish between the set of RIRs used for finding $\mathbf{L}$ and $\mathbf{R}$, denoted $\mathcal{H}_{\text{Model}}$, and the set of RIRs to be compressed, denoted $\mathcal{H}_{\text{Comp}}$. We will denote the cardinalities of these sets $n_{\text{Model}} = |\mathcal{H}_{\text{Model}}|$ and $n_{\text{Comp}} = |\mathcal{H}_{\text{Comp}}|$, respectively. To reflect the distinction between finding $\mathbf{L}$ and $\mathbf{R}$, summarized in Algorithm 1, and finding the matrices $\mathbf{D}_j$ for all the RIRs of $\mathcal{H}_{\text{Comp}}$, summarized in Algorithm 2, we will divide the original algorithm from [25], into two distinct algorithms. For Algorithm 1, we will initialize $\mathbf{L}$ as suggested in [25], by using

$$\mathbf{L}^{(0)} = \begin{bmatrix} \mathbf{I}_{\ell_1} \\ \mathbf{0} \end{bmatrix} \tag{4}$$

where $\mathbf{0}$ is a matrix of all zeroes, of appropriate size. As indicated in [25], the algorithm generally converges in very few iterations. We will therefore not consider a stopping criterion, but rather a maximum number of iterations, $I = 3$. It should be noted that when considering only one matrix, i.e. $n_{\text{Model}} = 1$ and by letting $\ell_1 = \ell_2 = R$, GLRAM is equivalent to an R-truncated SVD. The first proposed method, *Low-rank Compression of Room Impulse Responses for Low-latency Convolution* (LoCo-LoCo), consists of running Algorithm 1 and then Algorithm 2.

Algorithm 1 Finding $\mathbf{L}$ and $\mathbf{R}$

Input: $\{\mathbf{H}_j\}_{j\in\mathcal{H}_{\text{Model}}}$, $\mathbf{L}^{(0)}$, I
Output: $\mathbf{L}$, $\mathbf{R}$
for $i = 1, 2, \ldots, I$ **do**
  $\mathbf{M}_R = \sum_{j\in\mathcal{H}_{\text{Model}}} \mathbf{H}_j^T\mathbf{L}^{(i-1)}\mathbf{L}^{(i-1)^T}\mathbf{H}_j$
  Compute ℓ_2 first eigenvectors $\left\{\boldsymbol{\phi}_k^R\right\}_{k=1}^{\ell_2}$ of $\mathbf{M}_R$
  $\mathbf{R}^{(i)} \leftarrow \left[\boldsymbol{\phi}_1^R, \ldots, \boldsymbol{\phi}_{\ell_2}^R\right]$
  $\mathbf{M}_L = \sum_{j\in\mathcal{H}_{\text{Model}}} \mathbf{H}_j\mathbf{R}^{(i)}\mathbf{R}^{(i)^T}\mathbf{H}_j^T$
  Compute ℓ_1 first eigenvectors $\left\{\boldsymbol{\phi}_k^L\right\}_{k=1}^{\ell_1}$ of $\mathbf{M}_L$
  $\mathbf{L}^{(i)} \leftarrow \left[\boldsymbol{\phi}_1^L, \ldots, \boldsymbol{\phi}_{\ell_1}^L\right]$
end for
$\mathbf{L} \leftarrow \mathbf{L}^{(i)}$
$\mathbf{R} \leftarrow \mathbf{R}^{(i)}$

Algorithm 2 Finding $\mathbf{D}_j$

Input: $\{\mathbf{H}_j\}_{j\in\mathcal{H}_{\text{Comp}}}$, $\mathbf{L}$, $\mathbf{R}$
Output: $\{\mathbf{D}_j\}_{j=1}^{n_{\text{Comp}}}$
for $j = 1, \ldots, n_{\text{Comp}}$ **do**
  $\mathbf{D}_j \leftarrow \mathbf{L}^T\mathbf{H}_j\mathbf{R}$
end for

Next, the difference in spatial variability between the matrices $\mathbf{U}$ and $\mathbf{V}$ of an SVD of an RIR matrix $\mathbf{H}$, and by extension, the matrices $\mathbf{L}$ and $\mathbf{R}$ of the GLRAM, will be discussed. In preliminary simulations it was observed that for two separate SVDs of matricized RIRs corresponding to closely spaced receiver positions, the $\mathbf{V}$ matrices for the respective SVDs were much more similar than the respective $\mathbf{U}$ matrices. An example of this is displayed in Figure 1. There it can be seen that for three matricized RIRs, corresponding to closely spaced receiver positions, taken from [28], the columns of $\mathbf{U}$ (left) appear to change much more from one RIR to another, as compared to the columns of $\mathbf{V}$ (right). In Figure 1 we display the first (top) and second (bottom) columns of $\mathbf{U}$ and $\mathbf{V}$.

The merit of this will be further discussed in Section III-B. For now, it motives the proposal of the second method, *Low-rank Compression of Room Impulse Responses for Low-latency Convolution with Partially Invariant Transformation* (LoCo-LoCo-PIñaTa). This method consists of first finding the matrix $\mathbf{R}$ (common to all the compressed RIRs) by running Algorithm 1 (but ignoring $\mathbf{L}$), and then finding

$$\mathbf{W}_j = \mathbf{L}_j\mathbf{D}_j = \mathbf{L}_j\mathbf{L}_j^T\mathbf{H}_j\mathbf{R}, \tag{5}$$

$\mathbf{W}_j \in \mathbb{R}^{r\times\ell_2}$, by running Algorithm 3.

Algorithm 3 Finding $\mathbf{W}_j$

Input: $\{\mathbf{H}_j\}_{j\in\mathcal{H}_{\text{Comp}}}$, $\mathbf{R}$
Output: $\{\mathbf{W}_j\}_{j\in\mathcal{H}_{\text{Comp}}}$
for $j = 1, 2, \ldots, n_{\text{Comp}}$ **do**
  $\mathbf{M}_L = \mathbf{H}_j\mathbf{R}\mathbf{R}^T\mathbf{H}_j^T$
  Compute ℓ_1 first eigenvectors $\left\{\boldsymbol{\phi}_k^L\right\}_{k=1}^{\ell_1}$ of $\mathbf{M}_L$
  $\mathbf{L}_j \leftarrow \left[\boldsymbol{\phi}_1^L, \ldots, \boldsymbol{\phi}_{\ell_1}^L\right]$
  $\mathbf{W}_j = \mathbf{L}_j\mathbf{L}_j^T\mathbf{H}_j\mathbf{R}$
end for

D. COMPUTATIONAL COMPLEXITY AND COMPRESSION

The computationally most expensive steps of the GLRAM algorithm are the formation of the matrices $\mathbf{M}_R$ and $\mathbf{M}_L$, of $\mathcal{O}\left(\ell_1 c(r+c)n_{\text{Model}}\right)$ and $\mathcal{O}\left(\ell_2 r(r+c)n_{\text{Model}}\right)$, respectively [25], where $\mathcal{O}(\cdot)$ refers to the limiting number of multiplications. For GLRAM, and the algorithms considered here, the eigenvalue decomposition of $\mathbf{M}_R$ and $\mathbf{M}_L$, are also expensive, bounded at $\mathcal{O}(c^3)$ and $\mathcal{O}(r^3)$, respectively [29]. For Algorithms 1 and 3, the most expensive step therefore depends on the values of c, r, n_{Model}, and n_{Comp}. For Algorithm 2, the creation of $\mathbf{D}_j$ $j = 1, 2, \ldots, n_{\text{Comp}}$ is of order $\mathcal{O}\left(n_{\text{Comp}} r\ell_2(c+\ell_1)\right)$ [25].

With compression rate, we refer to the reduction in computational storage. For a single RIR, this is defined as

$$C(\hat{\mathbf{h}}) = 1 - \frac{\Upsilon(\hat{\mathbf{h}})}{n_h}, \tag{6}$$

where n_h is the number of coefficients of the recorded and truncated RIR and $\Upsilon(\hat{\mathbf{h}})$ is the number of coefficients for the compressed RIR. A compression rate close to zero means nearly no compression, whereas a compression rate closer to one means a high degree of compression. The benefit of GLRAM is that it allows us to consider $\mathbf{L}$ and/or $\mathbf{R}$ for the compression of multiple matricized RIRs simultaneously. In light of this, and denoting the number of $\mathbf{L}$ matrices by $n_{\mathbf{L}}$, the number of $\mathbf{R}$ matrices by $n_{\mathbf{R}}$, and we instead consider the compression rate for a set of n_{Comp} RIRs as

$$C(\hat{\mathbf{h}}) = 1 - \frac{n_{\mathbf{L}} r\ell_1 + n_{\text{Comp}}\ell_1\ell_2 + n_{\mathbf{R}} c\ell_2}{n_h n_{\text{Comp}}}. \tag{7}$$

When considering the scenario where $\mathbf{R}$ is common to all the RIRs of the room, but each RIR has its own $\mathbf{W}_j$, the expression for the compression rate is reduced to

$$C(\hat{\mathbf{h}}) = 1 - \frac{n_{\text{Comp}} r\ell_2 + c\ell_2}{n_h n_{\text{Comp}}}. \tag{8}$$

E. FAST LOW-LATENCY CONVOLUTION

How low-rank structure to the reshaped RIR can be leveraged for fast, low-latency convolution has been explored in previous work [22], [30]. Here we consider an input signal $\mathbf{x} \in \mathbb{R}^{n_x}$ and an output signal $\mathbf{y} = \mathbf{h} * \mathbf{x}$, $\mathbf{y} \in \mathbb{R}^{n_y}$, where $n_y = n_h + n_x - 1$. In the simplest case, where the RIR $\mathbf{h}$ is reshaped into a rank-1 matrix $\mathbf{H}$, the filtering operation with a very long filter (i.e., the RIR) is replaced by two filtering operations with significantly shorter filters, corresponding to the columns of $\mathbf{W}_j$ and $\mathbf{R}$. This unveils another benefit of a matrix $\mathbf{R}$ common to all the RIRs. The convolution between an audio signal and $\mathbf{R}$ needs to be done only once and the result can then be reused for all the considered RIRs. The extension to arbitrary rank of the matrix $\mathbf{H}$ is straightforward as a rank-R matrix is the sum of R rank-1 matrices. The filtering of the signal through the entire RIR can be replaced by the filtering through $2R$ shorter filters. The approach is summarized in Algorithm 4.[1] For Algorithm 4, we consider complexity in terms of number of multiply-add instructions. The creation of $\mathbf{P}$ requires $(n_x + (c-1)r)\ell_2(\lfloor\frac{n_x}{r}\rfloor + 1)$ multiply-add instructions and the creation of $\{\mathbf{y}_j\}_{j=1}^{n_{\text{Comp}}}$ requires in total $n_{\text{Comp}} n_y \ell_2 r$ multiply-add instructions. For most relevant scenarios $(n_x+(c-1)r)\ell_2(\lfloor\frac{n_x}{r}\rfloor+1) \ll n_{\text{Comp}} n_y \ell_2 r$. Traditional time-domain multi-channel convolution, assuming $n_h \geq n_x$, requires in total $n_{\text{Comp}} n_y n_h$ multiply-add instructions, hence the proposed approach yields a reduction of the computational cost by a factor of $\approx \frac{c}{\ell_2}$. A similar algorithm, exploiting both similar $\mathbf{L}$ and $\mathbf{R}$, could be envisaged, but as will be shown in Section III, LoCo-LoCo-PIñaTa is better suited for the purpose of compressing the RIRs of an entire room for low-latency convolution. For the sake of brevity, such an algorithm will therefore not be considered.

Algorithm 4 Multi-channel Low-rank Convolution

Input: $\{\mathbf{W}_j\}_{j\in\mathcal{H}_{\text{Comp}}}$, $\mathbf{R}$, $\mathbf{x}$
Output: $\{\mathbf{y}_j\}_{j=1}^{n_{\text{Comp}}}$
for $j = 1, \ldots, n_x + (c-1)r$ **do**
  for $\ell = 1, \ldots, \ell_2$ **do**
    $R_{\text{Low}} = \max\left(\lceil\frac{j-n_x}{r}\rceil + 1, 1\right)$
    $R_{\text{High}} = \min\left(\lceil\frac{j}{r}\rceil, c\right)$
    $x_{\text{Low}} = \max\left(\text{mod}(j-1, r) + 1, j - (c-1)r\right)$
    $x_{\text{High}} = \min\left(n_x - r + 1 + \text{mod}(j-1-n_x, r), j\right)$
    $\mathbf{P}[j, \ell] = \mathbf{R}[R_{\text{Low}} : R_{\text{High}}, \ell]\mathcal{I}\left(\mathbf{x}[x_{\text{Low}} : r : x_{\text{High}}]\right)$
  end for
end for
for $j = 1, \ldots, n_{\text{Comp}}$ **do**
  for $k = 1, \ldots, n_y$ **do**
    $P_{\text{Low}} = \max(k - r + 1, 1)$
    $P_{\text{High}} = \min(k, n_y - r + 1)$
    $W_{\text{Low}} = \max(k - (n_y - r), 1)$
    $W_{\text{High}} = \min(k, r)$
    for $\ell = 1, \ldots \ell_2$ **do**
      $\mathbf{p} = \mathbf{P}[P_{\text{Low}} : P_{\text{High}}, \ell]$
      $\mathbf{w} = \mathcal{I}\left(\mathbf{W}_j[W_{\text{Low}} : W_{\text{High}}, \ell]\right)$
      $\mathbf{y}_j[k] = \mathbf{y}_j[k] + \mathbf{w}^T\mathbf{p}$
    end for
  end for
end for

[1] A Matlab implementation of Algorithm 4 is available at https://github.com/m-jalmby/multichannel_conv_example

III. NUMERICAL RESULTS

A. COMPACT-ARRAY RIR MEASUREMENTS

In the first example we will be using RIRs from the *single- and multichannel audio recordings database* (SMARD) [28]. These are recorded at 48 kHz in a 60 m^2 room, with a reverberation time of approximately 0.15 s. The RIRs will be truncated at $n_h = 6400$ samples, corresponding to 0.13 seconds. We will consider the RIRs recorded with orthogonal arrays, which in turn consist of 3 uniform linear arrays (ULA) with 7 microphones (5 cm spacing) for each, recorded at 24 different source-receiver position configurations, yielding a total of 504 RIRs. In this scenario $\mathcal{H}_{\text{Model}} = \mathcal{H}_{\text{Comp}}$, i.e., the set of matrices used for finding $\mathbf{L}$ and $\mathbf{R}$ will be the same as the set of matrices to be compressed. We will consider two different versions of LoCo-LoCo and LoCo-LoCo-PIñaTa, respectively. For the first ones, which we denote LoCo-LoCo 7 and LoCo-LoCo-PIñaTa 7, we will consider a different $\mathcal{H}_{\text{Model}}$ for each ULA of the orthogonal array, i.e. $n_{\text{Model}} = 7$. For the other ones, LoCo-LoCo 21 and LoCo-LoCo-PIñaTa 21, $\mathcal{H}_{\text{Model}}$ is the entire orthogonal array, i.e. $n_{\text{Model}} = 21$. These two versions of LoCo-LoCo will be compared to two other approaches. Firstly, a low-rank approximation using a truncated SVD, for each RIR. The R-truncated SVD of a matrix $\mathbf{H}$ is the closest rank-R matrix to $\mathbf{H}$ in both the 2-norm and in the Frobenius norm [31]. The drawback from a compression point of view, however, is that for each compressed matricized RIR $\hat{\mathbf{H}}_j = \mathbf{U}_j[:, 1:R]\mathbf{S}_j[1:R, 1:R]\mathbf{V}_j[:, 1:R]^T$, the three matrices $\mathbf{U}_j$, $\mathbf{S}_j$, and $\mathbf{V}_j$ are unique to each approximated RIR $\hat{\mathbf{H}}_j$ and have to be stored. This means that for a fixed compression rate, the rank for the truncated-SVD approximation method will be significantly lower than ℓ_1 and ℓ_2 for LoCo-LoCo and LoCo-LoCo-PIñaTa. For the R-truncated SVD, $C(\hat{\mathbf{h}}) = 1 - R(c+r)/n_h$. For LoCo-LoCo, LoCo-LoCo-PIñaTa, and SVD, the truncated RIR vectors are reshaped into 80×80-matrices. We will also consider the state-of-the-art Opus interactive speech and audio codec [32], [33]. The Opus codec is created for the compression of audio, but has recently been considered for the compression of RIRs too [34]. The Opus encoding was done using Matlab's *audiowrite*. Although Opus shrinks the file size of the stored RIR, the number of coefficients remains the same. The error is measured in terms of normalized misalignment,

$$\mathcal{M}_{\text{dB}}\left(\hat{\mathbf{h}}\right) = 20\log_{10}\left(\frac{1}{n_{\text{Comp}}}\sum_{j=1}^{n_{\text{Comp}}}\frac{\|\hat{\mathbf{h}}_j - \mathbf{h}_j\|_2}{\|\mathbf{h}_j\|_2}\right). \quad (9)$$

The results are shown in Figure 2. There it can be seen that for the RIRs corresponding to the closely spaced microphones of SMARD, on either the full array or one for each ULA, LoCo-LoCo performs the best. LoCo-LoCo-PIñaTa, for which ℓ_1 and ℓ_2 is much smaller than for LoCo-LoCo, given a fixed compression rate, fails to perform on par with the benchmark methods, except for very high compression rates, where the performance of Opus declines considerably.

B. SPATIAL VARIATION OF $\mathbf{U}$ AND $\mathbf{V}$

The difference in how much of the spatial variation is reflected in $\mathbf{U}$ and $\mathbf{V}$, respectively, is highlighted in the following example. We consideri the same RIRs as in Section III-A and use an R-truncated SVD with $R = 12$. For each ULA of the orthogonal array, separate SVDs are made of the matricized RIRs, denote these $\mathbf{U}_n$ and $\mathbf{V}_n$, respectively, $n = 1, 2 \ldots, 7$. It is then explored what error is induced by using either $\mathbf{U}_1$ or $\mathbf{V}_1$, in place of $\mathbf{U}_n$ and $\mathbf{V}_n$, respectively, for the low-rank approximation. The averaged results are found in Figure 3 (top). For reference, using both $\mathbf{U}_n$ and $\mathbf{V}_n$ resulted in a normalized misalignment of ~ -18 dB. As expected, using $\mathbf{U}_n$ and $\mathbf{V}_n$ is by far the best, but the substantial outperformance by the case where $\mathbf{V}_1$ is kept constant, as compared to when $\mathbf{U}_1$ is kept constant, is an indication that more of the spatial invariance of adjacent RIRs of is captured by $\mathbf{V}$.

This is also seen when considering the subspace angles (see, e.g., [35]). For a matrix $\mathbf{U} \in \mathbb{R}^{m\times n}$, let $\mathcal{U} = \{\mathbf{U}[:, 1], \mathbf{U}[:, 2], \ldots, \mathbf{U}[:, n]\}$ denote the set of its column vectors, and U the subspace of $\mathbb{R}^m$ that the vectors of $\mathcal{U}$ span. For two matrices, $\mathbf{U} \in \mathbb{R}^{m\times n}, \mathbf{V} \in \mathbb{R}^{m\times p}$, the principal angles $\theta_1 \leq \theta_2 \leq \cdots \leq \theta_{\min(n,p)} \leq \pi/2$ between the two subspaces U and V and the corresponding principal directions $\mathbf{u}_k \in U$ and $\mathbf{v}_k \in V$ are defined recursively as

$$\begin{aligned} &\cos(\theta_k) = \mathbf{u}_k^T\mathbf{v}_k = \\ &\max_{\mathbf{u}\in U, \mathbf{v}\in V} \mathbf{u}^T\mathbf{v} \\ &\text{s.t. } \|\mathbf{u}\| = \|\mathbf{v}\| = 1 \\ &\mathbf{u}^T\mathbf{u}_i = \mathbf{v}^T\mathbf{v}_i = 0, i = 1, \ldots, k-1. \end{aligned} \quad (10)$$

In Figure 3 (bottom) we see the mean principal angles, averaged over the three ULA's of the 24 source-receiver position configurations, as a function of microphone position. The large degree of similarity between the subspaces V_1 and V_n, $n = 2, 3, \ldots, 7$, compared to U_1 and U_n, $n = 2, 3, \ldots, 7$, is evident. We will now consider how this can be exploited.

C. SYNTHETIC ROOM IMPULSE RESPONSES

We will here consider synthetically generated RIRs, in order to further investigate the spatial variation captured by $\mathbf{L}$ and $\mathbf{R}$, respectively. This allows for a scenario where the location of the loudspeaker generating the signal is fixed, and we consider the RIRs for microphones placed on a finely spaced grid throughout the room. The RIRs considered in this section have been generated using the image source method [36], [37]. The three-dimensional room is 3.7 m × 3.1 m × 3.2 m, the loudspeaker is placed at $\begin{bmatrix}2.62, & 1.4, & 1.6\end{bmatrix}$ m, and microphones are placed on a two-dimensional Cartesian grid with spacing 0.06 m, also at 1.6 m above the floor, starting 0.2 m from each wall. The frequencies of the excitation signal, the damping coefficients for the respective surfaces and frequencies, and the reverberation time, T_{60}, for the respective frequencies, are stated in Table 1. The

RIRs are truncated to 0.33 s, corresponding to $n_h = 15625$ samples, given the sampling rate of 48 kHz.

In this scenario we initially considered LoCo-LoCo and LoCo-LoCo-PIñaTa, and truncated SVDs as a baseline. For LoCo-LoCo, $\ell_1 = \ell_2 = 56$, for LoCo-LoCo-PIñaTa, $\ell_1 = \ell_2 = 21$, and for truncated SVD, $R = 12$, all corresponding to $C(\hat{\mathbf{h}}) \approx 0.8$, due to the fact that the number of coefficients shared by the compressed multi-channel RIRs varies strongly between the methods. For LoCo-LoCo and LoCo-LoCo-PIñaTa, the matrices common to all the compressed RIRs ($\mathbf{L}$ and $\mathbf{R}$ for LoCo-LoCo and $\mathbf{R}$ for LoCo-LoCo-PIñaTa) were found using a randomly chosen subset, $\mathcal{H}_{\text{Model}}$, with $n_{\text{Model}} = 25$, of the all RIRs to be compressed, $\mathcal{H}_{\text{Comp}}$, with $n_{\text{Comp}} = 2576$. Preliminary simulations showed poor performance by LoCo-LoCo. Despite the comparatively large ℓ_1 and ℓ_2, the matrices $\mathbf{L}$ and $\mathbf{R}$ are unable to capture the RIR variability throughout the room. Therefore, the results for LoCo-LoCo will not be shown here.

LoCo-LoCo-PIñaTa, on the other hand, is able to make use of the invariance of $\mathbf{R}$ throughout the room, while capturing the variability in $\mathbf{W}_j$. In order to be able to reliably represent RIRs from anywhere in the room, it is important to not only use RIRs that are far away from the source in $\mathcal{H}_{\text{Model}}$. If only RIRs too far from the source are used when modeling $\mathbf{R}$, the entries of the top row of $\mathbf{R}$ will be 0 which, in turn, will cause the first r taps of $\hat{\mathbf{h}}$ to be 0 as well. If then the direct component of an RIR in the room is at one of the first r samples, LoCo-LoCo-PIñaTa will be unable to faithfully restore the direct component, causing large misalignment.

The difference in normalized misalignment for LoCo-LoCo-PIñaTa and the SVD approximation is displayed in Figure 4. The locations at which the RIRs of $\mathcal{H}_{\text{Model}}$ for LoCo-LoCo-PIñaTa are sampled are marked in red. In Figure 4 it can be seen that LoCo-LoCo-PIñaTa performs better than the baseline, with the exception of a very small area in direct proximity of the source. For reference, the average normalized misalignment was -19.98 dB for LoCo-LoCo-PIñaTa and -18.73 dB for the baseline truncated SVD.

D. DISTRIBUTED-ARRAY RIR MEASUREMENTS

The findings from Section III-C are confirmed when applying the same methods to real data. In this section we consider RIRs from the dataset S32-M441 from [38]. Here, RIRs are recorded by microphones in a 1 m $\times$ 1 m planar grid, every 0.05 m, from $\{x, y | -0.5 \leq x, y \leq 0.5\}$, yielding 441 RIRs, with a source at $\begin{bmatrix} 1 & 0.45 & -0.1 \end{bmatrix}$, relative to the middle of the microphone grid. The RIRs are recorded in a room with approximate dimensions of 7.0 m $\times$ 6.4 m $\times$ 2.7 m, with $T_{60} = 0.19$ s. We consider $n_h = 8100$, corresponding to 0.17 s, and the RIRs are reshaped into 90×90 matrices. Much like in Section III-C, LoCo-LoCo fell short in preliminary simulations and will not be considered further. For LoCo-LoCo-PIñaTa, $\mathbf{R}$ was found using 40 randomly chosen RIRs. The average normalized misalignment is displayed in Figure 5, in the top plot as a function of compression rate and in the bottom plot as a function of convolution complexity, in terms of number of multiply-add instructions, when considering convolution with a signal of length $n_x = 100$. An example RIR, and the respective approximations, are displayed in Figure 6, zoomed in at samples $2800 - 4000$ in the time domain and the range $0 - 6000$ Hz in the frequency domain. LoCo-LoCo-PIñaTa's superior performance is evident in the time domain at the samples from 3200 to 3400, as it is able to capture a longer part of the original RIR. In the frequency domain, LoCo-LoCo-PIñaTa is able to better represent the dominant modal peaks.

There is a noticeable difference in improvement for using LoCo-LoCo-PIñaTa, as opposed to a traditional SVD, for the scenarios considered in Sections III-C and III-D. This is likely due to the fact that the synthetic RIRs of Section III-C display very little modal behavior, as compared to the recorded RIRs of Section III-D, as the low-rank property of a matrix such as the one in (2) is dependent on modal behavior, as described in (1). This is displayed in Figure 7, where the magnitude response of a measured RIR from [38] (top) and a synthetic RIR (bottom) are plotted. The magnitude of the synthetic RIR is dominated by the cavity mode at 0 Hz, whereas the measured RIR from [38] contains several distinct modal peaks.

E. CHANGES IN SOURCE POSITION

In this section we investigate how the different methods work for a scenario where the position of the source changes. The dataset S32-M441 from [38] that we will be using, contains RIR measurements from 441 different microphones and 32 different sources, for a total of $441 \cdot 32 = 14112$ RIRs. The source position is varied along two separate rectangles around the microphone grid, one placed 0.1 m above the microphones, the other placed 0.1 m below the microphones, each containing 16 source positions. Again we consider $n_h = 8100$, each truncated RIR is reshaped into a 90×90 matrix, and we let $\ell_1 = \ell_2 = 16$. We consider the difference in the quality of RIR approximation between two different versions of LoCo-LoCo-PIñaTa. In the first one, called *Common* $\mathbf{R}$, only one matrix $\mathbf{R}$ is found for each rectangle of sources, using 40 randomly selected RIRs out of the 441 RIRs measured for one of the 16 source positions. For the second one, called *Unique* $\mathbf{R}$, a separate $\mathbf{R}$ will be used for each source position, found using the same 40 randomly selected RIRs as for the first version. The compression rate is 0.82 in both cases. Averaged over all RIRs, the normalized misalignment is -23.95 dB for the Common $\mathbf{R}$ version and -23.96 dB for the Unique $\mathbf{R}$ version. The number of multiply-add instructions per output sample when considering convolution with a signal of length $n_x = 100$ is $1.44 \cdot 10^3$ in both scenarios. The main difference is the time spent retrieving $\mathbf{R}$. This needs to be done only twice for the Common $\mathbf{R}$ version, but 32 times for the Unique $\mathbf{R}$ version. Simulations were done using Matlab 2022b on a 2018 MacBook Pro with a 2.7

GHz QuadCore Intel Core i7 processor. Averaged over 50 Monte Carlo-simulations, the Common $\mathbf{R}$ version spent 0.18 s on the retrieval of $\mathbf{R}$ whereas Unique $\mathbf{R}$ spent 2.87 s. The results are summarized in Table 2. The minute difference in misalignment between the two considered versions indicates that the same matrix $\mathbf{R}$ can be used for the compression of multi-channel RIRs corresponding to multiple source positions throughout a room.

F. THE RELATIONSHIP BETWEEN ℓ_1 AND ℓ_2

Up until this point we have considered only the case where $\ell_1 = \ell_2$, where we remind the reader that ℓ_1 and ℓ_2 define the size of $\mathbf{D}_j \in \mathbb{R}^{\ell_1 \times \ell_2}$ in the approximated RIR $\hat{\mathbf{H}}_j = \mathbf{L}\mathbf{D}_j\mathbf{R}^T$. One of the advantages of the GLRAM, as compared to the SVD is that ℓ_1 and ℓ_2 can be chosen independently from each other. The previously established higher degree of similarity of the $\mathbf{R}$ matrices, as compared to the $\mathbf{L}$ matrices, for RIRs corresponding to adjacent receiver positions is an indication that $\ell_1 = \ell_2$ might not necessarily be the best choice. To demonstrate this, using the RIRs from [28] as described in Section III-A, we vary ℓ_1 between 5 and 35, while $\ell_2 = 40 - \ell_1$. The results are shown in Figure 8. In order to be able to illustrate both normalized misalignment and compression in the same figure, the y-axis in Figure 8 is average normalized misalignment per compression rate. Noticeable in Figure 8 is that the minimum does not occur at $\ell_1 = \ell_2 = 20$, but rather at $\ell_1 = 25$, $\ell_2 = 15$. This further strengthens the conclusion that it is favorable to yield more modeling capacity to $\mathbf{L}$, given that it absorbs more of the spatial variation. This is beneficial when considering the multi-channel low-rank convolution in Algorithm 4. As previously concluded, with a common $\mathbf{R} \in \mathbb{R}^{c \times \ell_2}$, the matrices $\mathbf{L}_j \in \mathbb{R}^{r \times \ell_1}$ and $\mathbf{D}_j \in \mathbb{R}^{\ell_1 \times \ell_2}$ can be stored together as $\mathbf{W}_j \in \mathbb{R}^{r \times \ell_2}$. A smaller ℓ_2 decreases the overall complexity of the convolution.

IV. CONCLUSIONS

In this paper propose two novel methods for the joint compression of multiple RIRs by use of joint low-rank approximations. The first proposed method, LoCo-LoCo, proved better than the benchmark methods, truncated SVD and state-of-the-art audio compression standard Opus, in scenarios where the RIRs to be approximated correspond to very closely spaced receivers. In scenarios where the receivers are farther apart, the other proposed method, LoCo-LoCo-PIñaTa, outperformed the benchmark methods, exploiting the demonstrated spatial invariability in one of the components of the GLRAM decomposition used throughout this paper. The compressed multi-channel RIRs yielded by the proposed methods are amenable to fast low-latency multi-channel convolution and for multi-channel RIRs compressed with LoCo-LoCo-PIñaTa we provide an explicit convolution algorithm.

Previous research has revealed a significant improvement in performance when considering a 3D tensor approximation of a single RIR, as opposed to a 2D matrix approximation, given a fixed compression rate. This gives reason to believe that tensor approximations could lead to an improvement of the joint compression of multiple RIRs considered in this paper and should be the focus of future research.

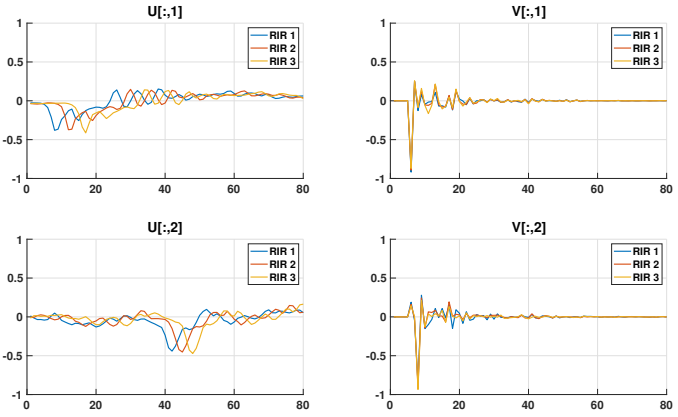

FIGURE 1. Similarity of SVD for closely spaced RIRs. Columns of $\mathbf{U}$ (left) and $\mathbf{V}$ (right). First column (top), second column (bottom).

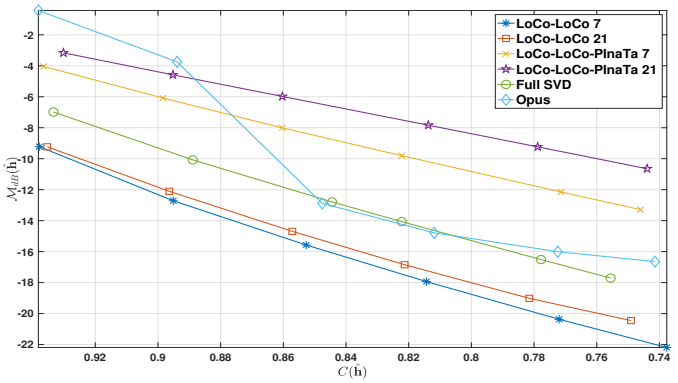

FIGURE 2. Normalized misalignment, $\mathcal{H}_{\text{Model}} = \mathcal{H}_{\text{Comp}}$

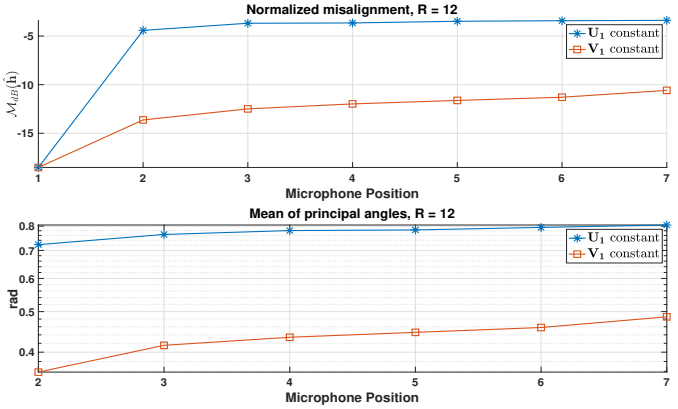

FIGURE 3. Spatial variation of $\mathbf{U}$ and $\mathbf{V}$ (top) and mean of principal angels between subspaces spanned by the columns of $\mathbf{U}_1$ and $\mathbf{U}_n$, and $\mathbf{V}_1$ and $\mathbf{V}_n$, respectively

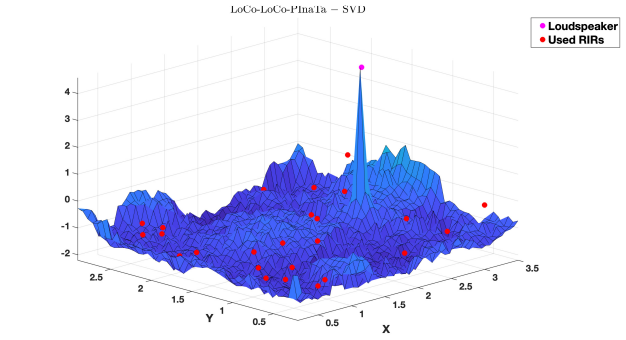

FIGURE 4. Difference between normalized misalignment for LoCo-LoCo-PIñaTa and normalized misalignment for SVD

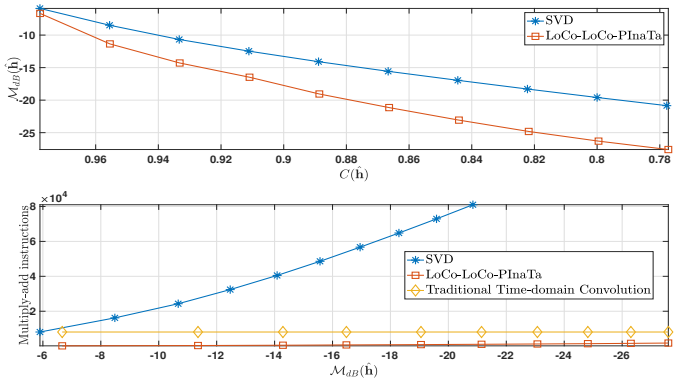

FIGURE 5. Normalized misalignment for distributed-array RIR measurements

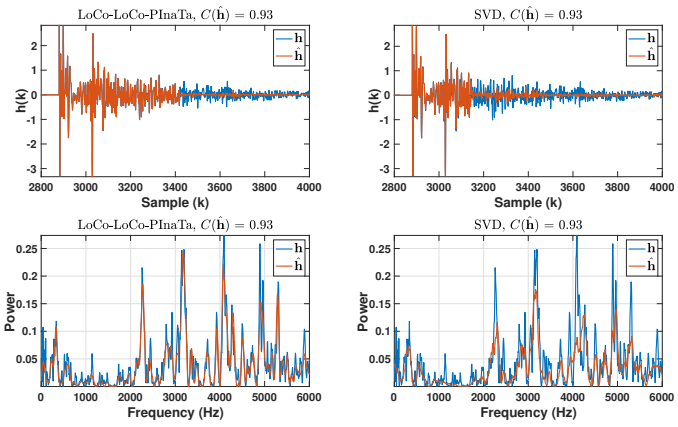

FIGURE 6. Example RIR from [38] and compressed RIRs obtained with LoCo-LoCo-PIñaTa (left) and SVD (right), in the time domain (top) and in the frequency domain (bottom).

TABLE 1. Damping coefficients and reverberation times, Section III-C

Surface / Hz	125	250	500	1000	2000	4000
Walls	0.1	0.2	0.4	0.6	0.5	0.6
Ceiling	0.02	0.02	0.03	0.03	0.04	0.07
Floor	0.02	0.02	0.03	0.03	0.04	0.07
T_{60}	1.23	0.63	0.33	0.22	0.26	0.21

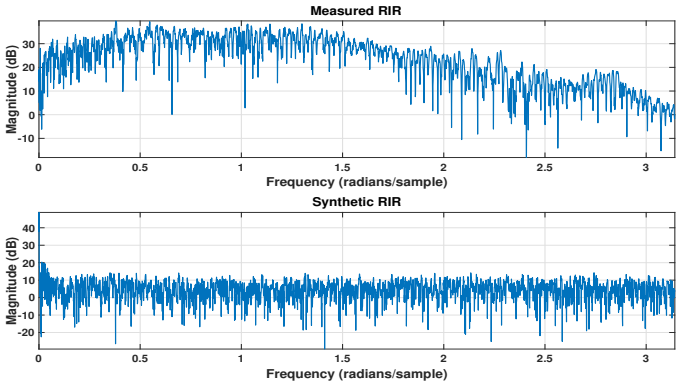

FIGURE 7. Example RIR from [38] (top) and synthetic RIR (bottom)

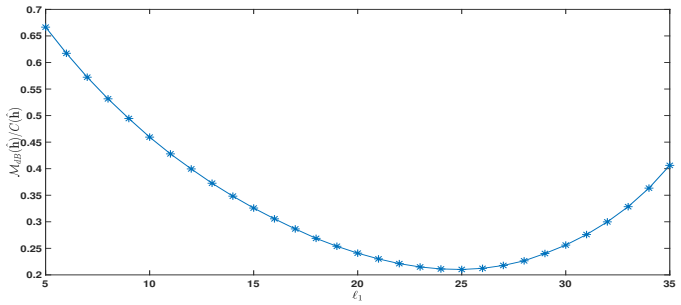

FIGURE 8. Normalized misalignment per compression rate, for varying ℓ_1 and ℓ_2.

TABLE 2. Changes in source position

Measure	Common $\mathbf{R}$	Unique $\mathbf{R}$
$C(\hat{\mathbf{h}})$	0.82	0.82
$\ell_1 = \ell_2$	16	16
$\mathcal{M}_{\text{dB}}(\hat{\mathbf{h}})$	-23.95	-23.96
Multiply-add instructions per output sample	$1.44 \cdot 10^3$	$1.44 \cdot 10^3$
Retrieval of $\mathbf{R}$	0.18 s	2.87 s

REFERENCES

[1] C. Schissler, P. Stirling, and R. Mehra, "Efficient construction of the spatial room impulse response," in *Proc. 2017 IEEE Virtual Reality (VR) Conf.*, 2017, pp. 122–130.

[2] C. Evers, H. W. Löllmann, H. Mellmann, A. Schmidt, H. Barfuss, P. A. Naylor, and W. Kellermann, "The LOCATA challenge: Acoustic source localization and tracking," *IEEE/ACM Trans. Audio Speech Lang. Process.*, vol. 28, pp. 1620–1643, 2020.

[3] S. Gannot, E. Vincent, S. Markovich-Golan, and A. Ozerov, "A consolidated perspective on multimicrophone speech enhancement and source separation," *IEEE/ACM Trans. Audio Speech Lang. Process.*, vol. 25, no. 4, pp. 692–730, 2017.

[4] J. Mourjopoulos and M. Paraskevas, "Pole and zero modeling of room transfer functions," *J. Sound Vib.*, vol. 146, no. 2, pp. 281–302, 1991.

[5] G. Vairetti, E. De Sena, M. Catrysse, S. H. Jensen, M. Moonen, and T. van Waterschoot, "A scalable algorithm for physically motivated and sparse approximation of room impulse responses with orthonormal basis functions," *IEEE/ACM Trans. Audio Speech Lang. Process.*, vol. 25, no. 7, pp. 1547–1561, 2017.

[6] C. Huszty, N. Bukuli, Á. Torma, and F. Augusztinovicz, "Effects of filtering of room impulse responses on room acoustics parameters by using different filter structures," *J. Acoust. Soc. Amer.*, vol. 123, p. 3617, 2008.

[7] G. Vairetti, "Efficient parametric modeling, identification and equalization of room acoustics," Ph.D. dissertation, KU Leuven, 2018.

[8] L. S. H. Ngia, "Recursive identification of acoustic echo systems using orthonormal basis functions," *IEEE Trans. Speech Audio Process.*, vol. 11, no. 3, pp. 278–293, 2003.

[9] P. A. Naylor and N. D. Gaubitch, *Speech Dereverberation*. Springer, 2010.

[10] T. Rossing, *Springer Handbook of Acoustics*. Springer, 2014.

[11] K. Shi, X. Ma, and G. Tong Zhou, "An efficient acoustic echo cancellation design for systems with long room impulses and nonlinear loudspeakers," *Signal Processing*, vol. 89, no. 2, pp. 121–132, 2009.

[12] L. Krishnan, P. D. Teal, and T. Betlehem, "A robust sparse approach to acoustic impulse response shaping," in *Proc. 2015 IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, 2015, pp. 738–742.

[13] H. Hacihabiboglu, E. De Sena, Z. Cvetkovic, J. Johnston, and J. O. Smith III, "Perceptual spatial audio recording, simulation, and rendering: An overview of spatial-audio techniques based on psychoacoustics," *IEEE Signal Process. Mag.*, vol. 34, no. 3, pp. 36–54, 2017.

[14] T. Ajdler, L. Sbaiz, and M. Vetterli, "The plenacoustic function and its sampling," *IEEE Trans. Signal Process.*, vol. 54, no. 10, pp. 3790–3804, 2006.

[15] I. Wohlgenannt, A. Simons, and S. Stieglitz, "Virtual reality," *Business & Information Systems Engineering*, vol. 62, no. 5, pp. 455–461, 2020.

[16] H. S. Llopis, F. Pind, and C.-H. Jeong, "Development of an auditory virtual reality system based on pre-computed b-format impulse responses for building design evaluation," *Building and Environment*, vol. 169, 2020.

[17] B. Xie, H. Liu, R. Alghofaili, Y. Zhang, Y. Jiang, F. D. Lobo, C. Li, W. Li, H. Huang, M. Akdere, C. Mousas, and L. Yu, "A review on virtual reality skill training applications," *Frontiers in Virtual Reality*, vol. 2, 2021.

[18] P. M. Emmelkamp and K. Meyerbröker, "Virtual reality therapy in mental health," *Annual Review of Clinical Psychology*, vol. 17, no. 1, pp. 495–519, 2021.

[19] J. Zhao, X. Zheng, C. Ritz, and D. Jang, "Interpolating the directional room impulse response for dynamic spatial audio reproduction," *Applied Sciences*, vol. 12, no. 4, 2022.

[20] M. Jälmby, F. Elvander, and T. van Waterschoot, "Low-rank tensor modeling of room impulse responses," in *Proc. 29th European Signal Process. Conf. (EUSIPCO)*, 2021, pp. 111–115.

[21] ——, "Low-rank room impulse response estimation," *IEEE/ACM Trans. Audio Speech Lang. Process.*, vol. 31, pp. 957–969, 2023.

[22] ——, "Fast low-latency convolution by low-rank tensor approximation," in *Proc. IEEE Int. Conf. on Acoust., Speech Signal Process. (ICASSP)*, 2023, pp. 1–5.

[23] ——, "Compression of room impulse responses for compact storage and fast low-latency convolution," KU Leuven ESAT-STADIUS Techical Report 23-149, 2023, https://ftp.esat.kuleuven.be/pub/stadius/mjalmby/23-149.pdf.

[24] J. Brunnström, M. Jälmby, T. van Waterschoot, and M. Moonen, "Fast low-rank filtered-x least mean squares for multichannel active noise control," accepted for publication in Proc. 2023 Asilomar Conf. Signals Syst. Comput., 2023, https://ftp.esat.kuleuven.be/pub/stadius/jbrunnst/2023_asilomar_low_rank_fxlms_006.pdf.

[25] J. Ye, "Generalized low rank approximations of matrices," in *Proc. Twenty-First Int. Conf. Machine Learning*, 2004, pp. 112–119.

[26] G. Huang, J. Benesty, J. Chen, C. Paleologu, S. Ciochină, W. Kellermann, and I. Cohen, "Acoustic system identification with partially time-varying models based on tensor decompositions," in *Proc. 2022 Int. Workshop Acoustic Signal Enhancement (IWAENC)*, 2022, pp. 1–5.

[27] M. Boussé, O. Debals, and L. De Lathauwer, "A tensor-based method for large-scale blind source separation using segmentation," *IEEE Trans. Signal Process.*, vol. 65, no. 2, pp. 346–358, 2017.

[28] J. K. Nielsen, J. R. Jensen, S. H. Jensen, and M. G. Christensen, "The single- and multichannel audio recordings database (SMARD)," in *Proc. 2014 Int. Workshop Acoustic Signal Enhancement (IWAENC)*, 2014, pp. 40–44.

[29] V. Y. Pan and Z. Q. Chen, "The complexity of the matrix eigenproblem," in *Proc. of the Thirty-First Annual ACM Symposium on Theory of Computing*, 1999, pp. 507–516.

[30] J. Atkins, A. Strauss, and C. Zhang, "Approximate convolution using partitioned truncated singular value decomposition filtering," in *Proc. 2013 IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2013, pp. 176–180.

[31] G. H. Golub and C. F. Van Loan, *Matrix computations*. JHU press, 2013.

[32] J.-M. Valin, K. Vos, and T. Terriberry, *Definition of the Opus audio codec*, IETF, September 2012.

[33] J.-M. Valin, G. Maxwell, T. B. Terriberry, and K. Vos, "High-quality, low-delay music coding in the Opus codec," *J. Audio Eng. Soc.*, 2013.

[34] H. Ren, C. Ritz, J. Zhao, and D. Jang, "Impact of compression on the performance of the room impulse response interpolation approach to spatial audio synthesis," in *Proc. 2022 Asia-Pacific Signal and Information Process. Association Annual Summit and Conf. (APSIPA ASC)*, 2022, pp. 442–448.

[35] P. Van Overschee and B. De Moor, *Subspace identification for linear systems: Theory — Implementation — Applications*. Springer, 2012.

[36] J. B. Allen and D. A. Berkley, "Image method for efficiently simulating small-room acoustics," *Acoust. Soc. of Amer.*, vol. 65, no. 4, pp. 943–950, 1979.

[37] Mathworks, "Room impulse response simulation with the image-source method and hrtf interpolation," https://nl.mathworks.com/help/audio/ug/room-impulse-response-simulation-with-image-source-method-and-hrtf-interpolation.html.

[38] S. Koyama, T. Nishida, K. Kimura, T. Abe, N. Ueno, and J. Brunnström, "MESHRIR: A dataset of room impulse responses on meshed grid points for evaluating sound field analysis and synthesis methods," in *Proc. 2021 IEEE Workshop Appl. of Signal Process. Audio Acoust. (WASPAA)*, 2021, pp. 1–5.