Spatial Acoustics Library for MATLAB (SALM): A Computational Toolkit for Spatial Audio Processing

César D. Salvador (1), Jorge Treviño (2) and Shuichi Sakamoto (2)

(1) Research Department and School of Electronic Engineering, Faculty of Engineering, Universidad Peruana de Ciencias Aplicadas (UPC), Lima, Peru, cesar.salvador@upc.edu.pe

(2) Research Institute of Electrical Communication (RIEC) and Graduate School of Information Sciences (GSIS), Tohoku University, Sendai, Japan, jorge.trevino@music.yamaha.com and saka@ais.riec.tohoku.ac.jp

* Jorge Treviño is currently affiliated with Yamaha Corporation

** SALM is available at: https://github.com/cesardsalvador/SpatialAcousticsLibraryMATLAB

Outline

1. Introduction
2. Structure of SALM
3. Geometry
4. Use Case 1: Diffuse-Field Equalization of HRTFs
5. Use Case 2: Distance Extrapolation of HRTFs
6. Future Work
7. Conclusion

1. Introduction

- Spatial audio is central to VR, AR, binaural rendering and room acoustics
- Existing MATLAB/Octave libraries for spatial audio:
  - SOFiA (B. Bernschutz *et al.*, 2011)
  - SFS Toolbox (H. Wierstorf and S. Spors, 2012)
  - Aktools (F. Brinkmann and S. Weinzierl, 2017)
  - ITA Toolbox (M. Berzborn *et al.*, 2017)
- Motivation of SALM
  - Each library brings unique strengths and has advanced spatial audio functions
  - SALM complements these efforts by offering a unified framework
  - Emphasis on transform-domain tools and reproducible workflows

1. Introduction – Existing MATLAB/Octave Libraries for Spatial Audio

Library	Description	Spatial-domain processing	Transforms	Transform-domain processing
SOFiA Toolbox [13]	MATLAB library to analyze a sound field captured with a microphone array.	Microphone-array handling; spherical-grid management; visualization of measured fields.	Spherical Fourier transforms.	Modal beamforming; plane-wave decomposition; radial filtering for rigid/open spheres.
Sound field synthesis toolbox [14]	MATLAB/Octave library to synthesize a sound field in an area surrounded by a loudspeaker array.	Driving functions for loudspeaker-array rendering; convolution engine.	Spherical Fourier transforms.	Modal driving filters; binaural rendering from arrays.
AKtools [15]	MATLAB library for the capture, processing, analysis and rendering of spatial audio signals.	Signal generation; convolution engine; room analysis.	Spherical Fourier transforms.	Radial filtering; binaural rendering.
ITA Toolbox [16]	MATLAB toolbox for acoustic measurements and audio signal processing.	Impulse-response measurement; convolution engines for auralization.	Spherical Fourier transforms.	Radial filtering; modal analysis.
SALM	MATLAB library for sound field analysis, processing and synthesis with circular and spherical arrays.	Diffuse-field filter for equalization; free-field translation operator for acoustic centering.	Circular, semicircular, and spherical Fourier transforms.	Distance-varying filter (DVF) for radial extrapolation; boundary-matching filter (BMF) for array signal conversion.

2. Structure of SALM

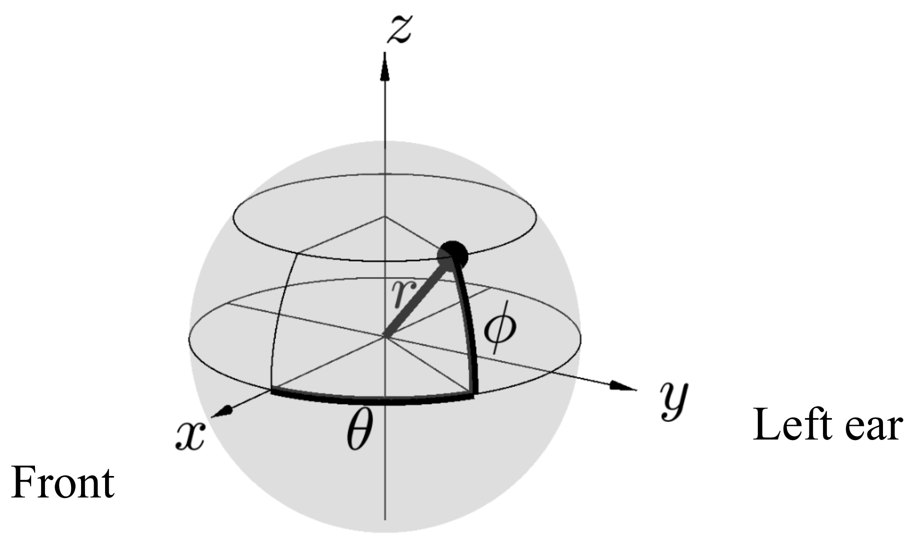
Special functions
`pnm`: associated Legendre polynomial
`ynm`: spherical harmonics
`besseljsph`: spherical Bessel function
`besselysph`: spherical Neumann function
`besselhsph`: spherical Hankel function
`dbesseljsph`: derivative of `besseljsph`
`dbesselhsph`: derivative of `besselhsph`

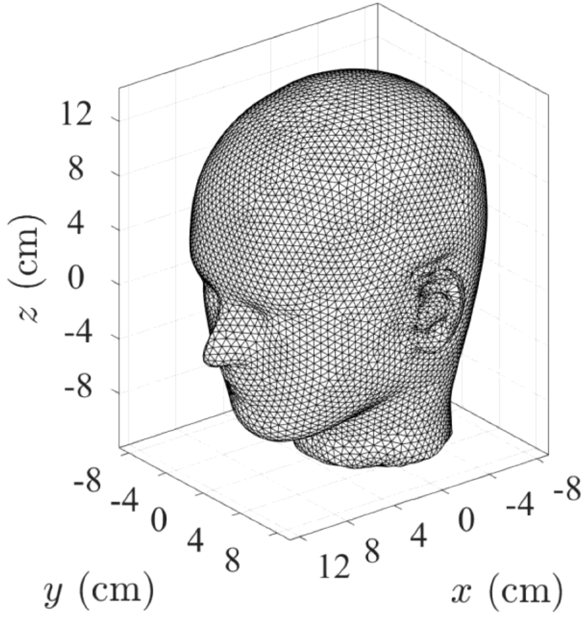
Spatial-domain functions
`diffuseFieldFilter`
`freeFieldtranslationOperator`

Transform functions
`cft`: circular Fourier transform
`icft`: inverse circular Fourier transform
`flt`: semicircular Fourier-Legendre transform
`iflt`: inverse semicircular Fourier-Legendre transform
`sft`: spherical Fourier transform
`isft`: inverse spherical Fourier transform
`pinvreg`: Tikhonov regularized pseudoinverse

Transform-domain functions
`dvf`: distance-varying filters
`bmf`: boundary-matching filters

3. Geometry





$$\vec{r} = (r, \theta, \phi)$$

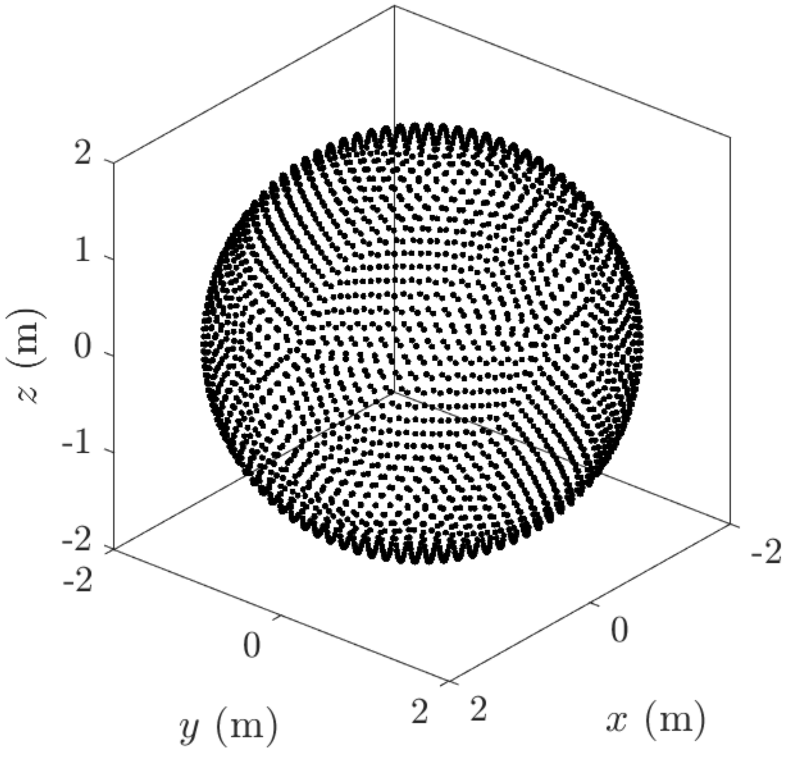
radial distance r

azimuthal angle $\theta \in [-\pi, \pi]$

elevation angle $\phi \in [-\frac{\pi}{2}, \frac{\pi}{2}]$

Interaural-polar spherical coordinates are also supported. See `cart2isph` and `isph2cart`.

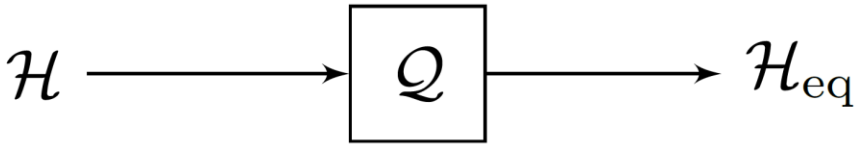
4. Use Case 1: Diffuse-Field Equalization of HRTFs



$$\{\mathcal{H}_\ell\}, \ell = 1, \ldots, L,$$

where L is the number of directions.

`diffuseFieldFilter`

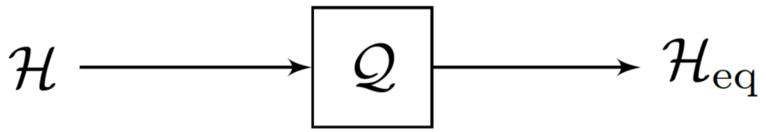


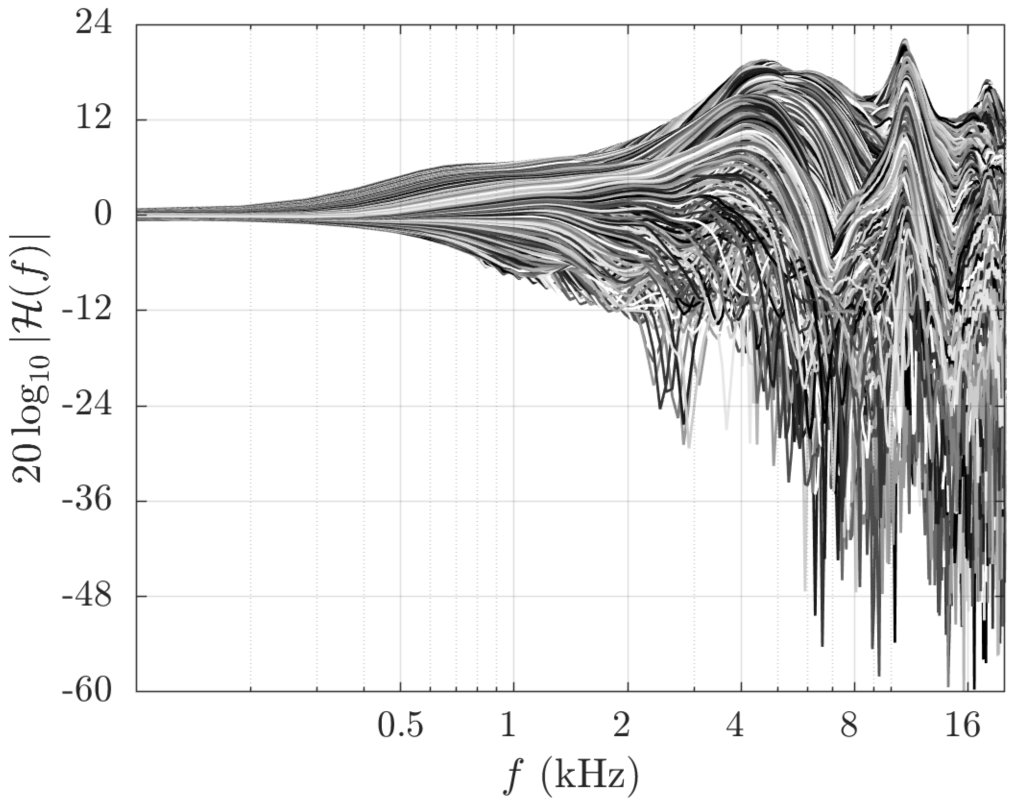
$$\mathcal{Q} = |\mathcal{Q}| \exp(j\angle\mathcal{Q})$$

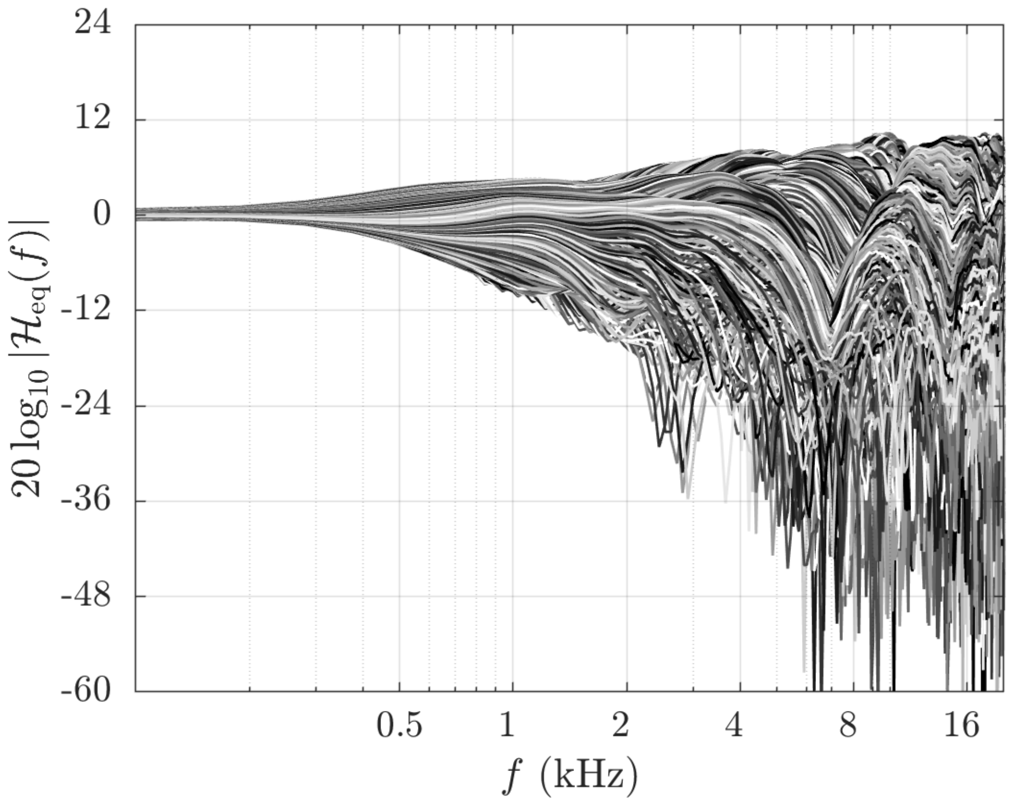
$$|\mathcal{Q}| = \left(\frac{1}{\sum_{\ell=1}^{L} |\mathcal{H}_\ell|^2 w_\ell} \right)^{\frac{1}{2}}$$

$$\angle\mathcal{Q} = \begin{cases} 0, & \text{zero-phase,} \\ \mathfrak{H}\left\{\log \frac{1}{|\mathcal{Q}|}\right\}, & \text{minimum-phase,} \end{cases}$$

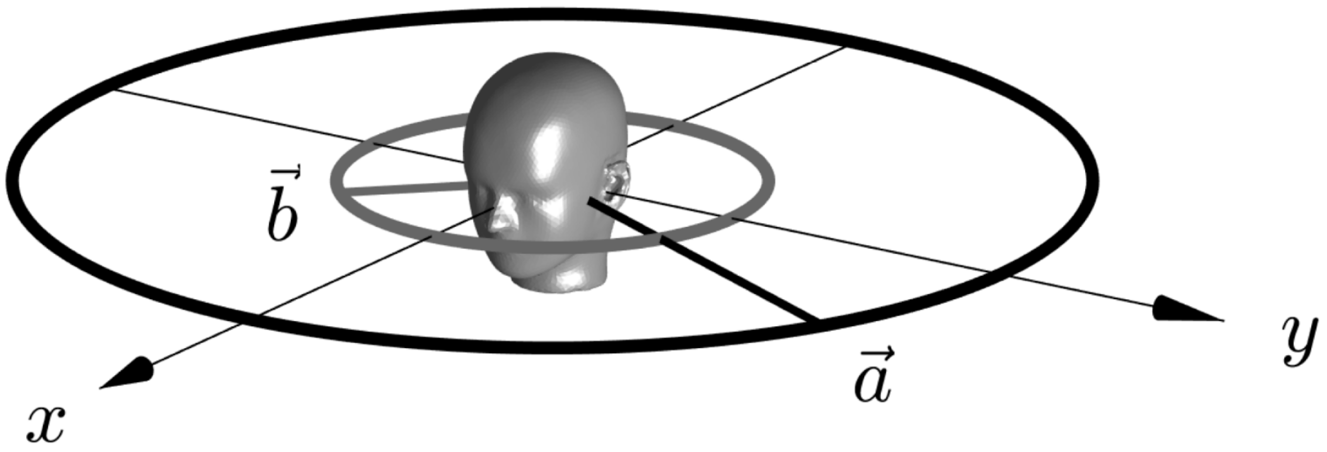
4. Use Case 1: Diffuse-Field Equalization of HRTFs



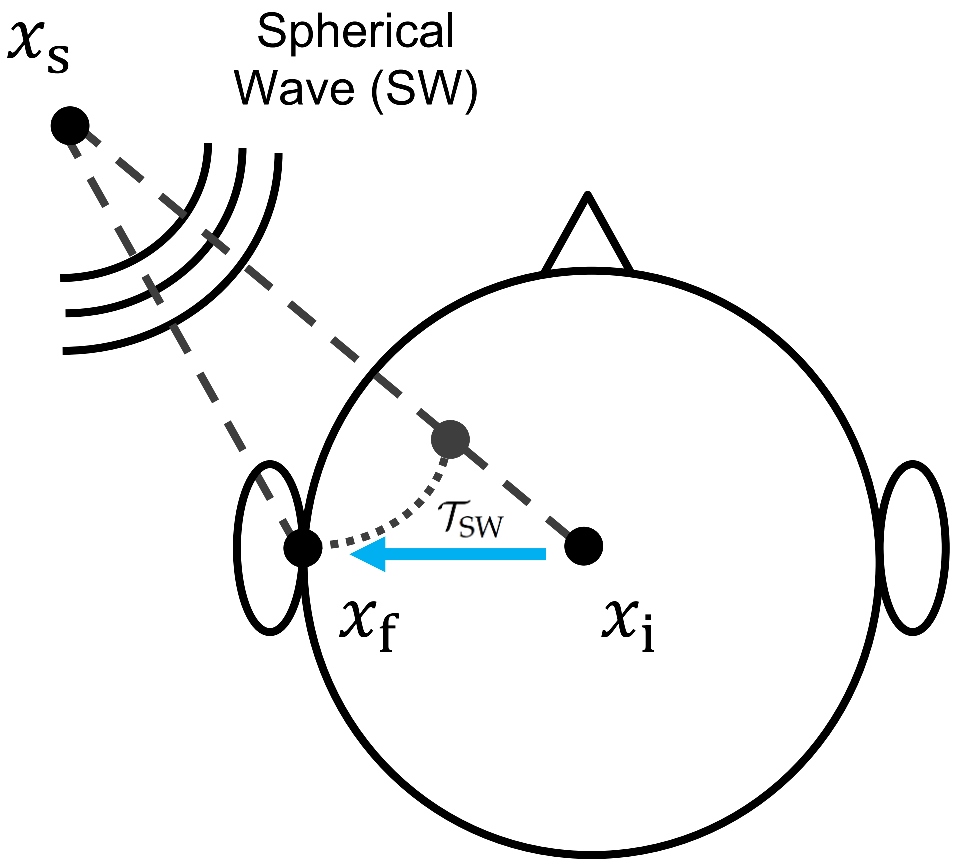




5. Use Case 2: Distance Extrapolation of HRTFs – Geometry



5. Use Case 2: Distance Extrapolation of HRTFs – Ear Centering



`freeFieldTranslationOperator`

$$\mathcal{T}_{\mathrm{SW}}(\vec{x}_\mathrm{i}, \vec{x}_\mathrm{f}) = \frac{\|\vec{x}_\mathrm{s} - \vec{x}_\mathrm{i}\|}{\|\vec{x}_\mathrm{s} - \vec{x}_\mathrm{f}\|} e^{jk(\|\vec{x}_\mathrm{s}-\vec{x}_\mathrm{i}\| - \|\vec{x}_\mathrm{s}-\vec{x}_\mathrm{f}\|)}$$

5. Use Case 2: Distance Extrapolation of HRTFs

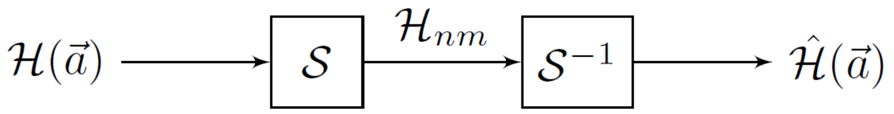


Fig. 5. Interpolation along direction using the direct and inverse spherical Fourier transforms $\mathcal{S}$ and $\mathcal{S}^{-1}$.

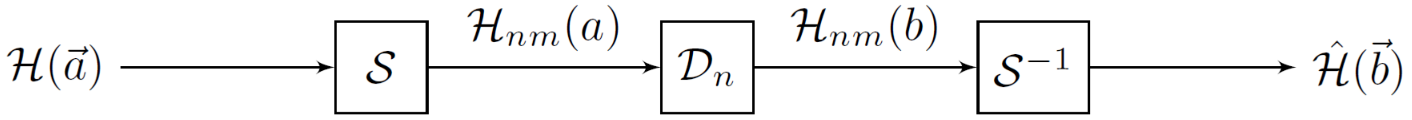


Fig. 6. Extrapolation along distance using the distance-varying filter $\mathcal{D}_n$.

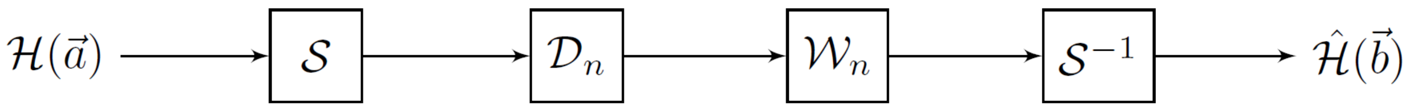


Fig. 7. Regularized extrapolation along distance using the window $\mathcal{W}_n$.

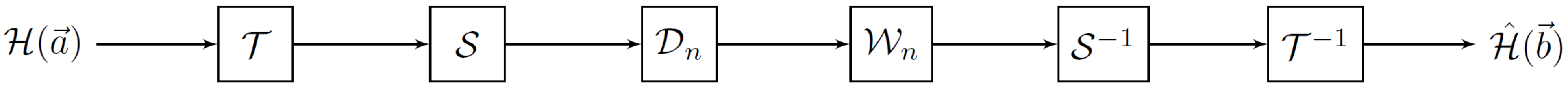
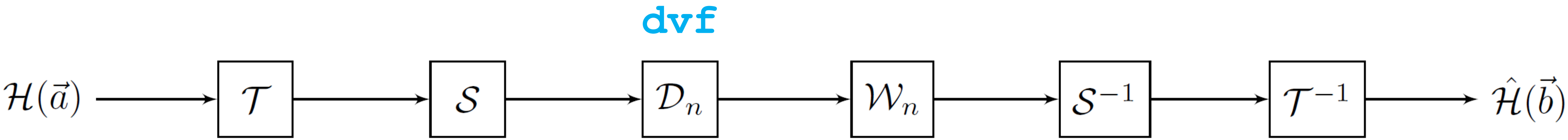
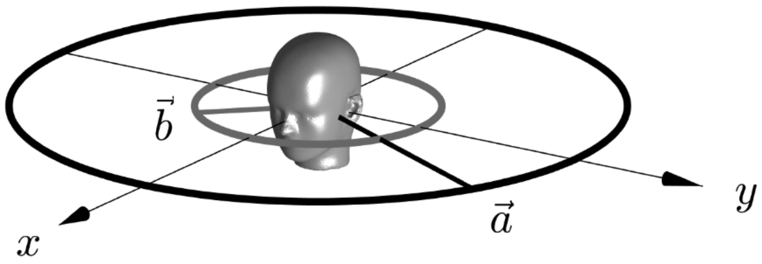


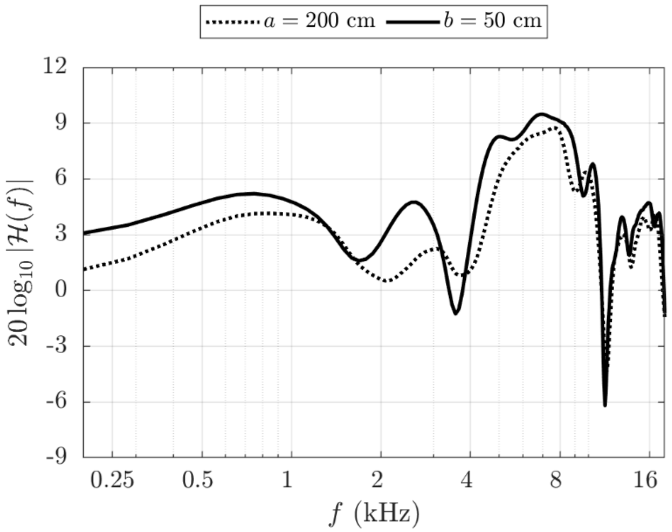
Fig. 8. Ear-centered extrapolation along distance using the direct and inverse translators $\mathcal{T}$ and $\mathcal{T}^{-1}$.

5. Use Case 2: Distance Extrapolation of HRTFs



$$\mathcal{D}_n(a,b) = \frac{h_n^{(i)}(kb)}{h_n^{(i)}(ka)} \qquad \mathcal{W}_n = \frac{1}{1 + \left(\frac{b}{a}\right)^2 |\mathcal{D}_n|^2}$$





6. Future Work

- Benchmarking with large SOFA datasets
- Extending to:
  - Circular DVFs
  - Spatial metrics for clarity
- Python port for wider accessibility: SALP
- Integration with perceptual testing frameworks

7. Conclusion

- SALM = unified, extensible, reproducible
- Bridges theory ↔ applications
- Applications: binaural rendering and architectural acoustics
- Contribution to the 3D audio research community

Thanks!

César D. Salvador, Jorge Treviño and Shuichi Sakamoto

SALM is available at: https://github.com/cesardsalvador/SpatialAcousticsLibraryMATLAB