Updates on the Real-Time Spherical Array Renderer (ReTiSAR)

Hannes Helmholz[1], Tim Lübeck[2,3], Jens Ahrens[1],
Sebastià V. Amengual Garí[4], David Lou Alon[4] and Ravish Mehra[4]

[1] *Chalmers University of Technology, Gothenburg, Sweden*
[2] *TH Köln - University of Applied Sciences, Cologne, Germany*
[3] *Technical University of Berlin, Berlin, Germany*
[4] *Facebook Reality Labs, Redmond, USA*
Email: hannes.helmholz@chalmers.se

Abstract

We recently presented ReTiSAR, a framework for binaural rendering of spherical microphone array signals in real-time. The array signals and the employed head-related transfer functions are convolved in the spherical harmonics domain to compute the resulting ear signals and virtually place a listener into the captured sound field. In this contribution, we present the latest additions to the Python software package. These comprise, among others, an interface to the Spatially Oriented Format for Acoustics (SOFA), the ability of switching the spherical harmonics rendering order during runtime as well as performance optimizations enabling real-time rendering up to 12th order. Furthermore, we integrated enhancements recently proposed in the literature for the upper frequency range where spatial undersampling occurs.

Introduction

Spherical microphone arrays (SMAs) are an established tool for the accurate capture and reproduction of spatial sound fields. The array signals can be auralized over headphones for a single listener by means of a processing pipeline, such as the recently presented *Real-Time Spherical Array Renderer* (ReTiSAR) [7]. With the code publicly available[1], we refined all stages of the Python implementation by further developing the rendering, performance, and convenience functionality.

ReTiSAR enables the capture of signals from a physical SMA, allowing for a live-rendition of the surrounding space with minimal delay. Furthermore, the pipeline can auralize measured high-resolution data sets of array room impulse responses (ARIRs) (e.g. [14]) with arbitrary source material in real-time. The renderer was also utilized in a number of publications, investigating the propagation of microphone self-noise to the ear signals for different spherical sampling grids, rendering configurations and non-uniform noise contributions [5, 6].

Rendering Method

An incoming sound field can be captured with a SMA by sampling it at discrete sensor positions on the surface of a sphere. ReTiSAR realizes the binaural rendering entirely in the spherical harmonics (SH) domain [1], contrary to the otherwise popular *virtual loudspeaker* approach [3]. Fig. 1 visualizes a generalized signal flow of the SH rendering pipeline from SMA to the ear signals. The resulting left and right ear signals are calculated by

$$E_{\mathrm{L,R}}(\omega) = \sum_{n=0}^{N} \sum_{m=-n}^{n} \underbrace{(-1)^m\ d_n(\omega)\ \mathring{H}_n^m(\omega)}_{\mathring{B}_n^m(\omega)}\ \mathring{S}_n^{-m}(\omega)\ \mathrm{e}^{-\mathrm{j}m\alpha} \tag{1}$$

whereby $\mathring{S}_n^{-m}(\omega)$ denote the captured microphone signals transformed into the spherical harmonics (SH) domain by means of plane wave decomposition [12]. Depending on the configuration, ReTiSAR can realize this expansion into SH coefficients by means of discretization of the transformation integral, which requires knowledge of the *quadrature weights*, or by a (weighted) least-squares matrix inversion [13].

A similar transformation is applied to a set of head-related impulse responses (HRIRs) to compute $\mathring{H}_n^m(\omega)$. Such data sets contain generic artificial or individual head models, in the following being virtually exposed to the captured sound field.

Also involved in the rendering process is a set of *modal radial filters* (MRF) $d_n(\omega)$ that compensate for the spatial extent and properties of the specific SMA scattering body. The filters exhibit therefore properties that are dependent on the array radius and the SH processing order. Theoretically, these filters exhibit very large amplification gains at low and high frequencies, which we restrict by means of a soft-clipping approach [4, 3].

In an ideal case, the output of the binaural rendering pipeline would be identical to a listener's ear signals being virtually located in the sound field at the position of the SMA. The rendering accounts for the instantaneous head orientation in the SH domain by rotating the HRIRs relative to the sound field. ReTiSAR is currently limited to head rotations around the vertical axis as represented by an angle α in the factor $\mathrm{e}^{-\mathrm{j}m\alpha}$ in Eq. 1.

All static parts of the processing are pre-computed during initialization of the pipeline in order to optimize the rendering performance [7]. As shown by $\mathring{B}_n^m(\omega)$ in Eq. 1 and highlighted grey in Fig. 1, the static components comprise the modal radial filters as well as the transformed head model. Also included are any potentially applied mitigation techniques for errors due to a deviation from the theoretic requirements as presented below.

[1] `https://github.com/AppliedAcousticsChalmers/ReTiSAR`

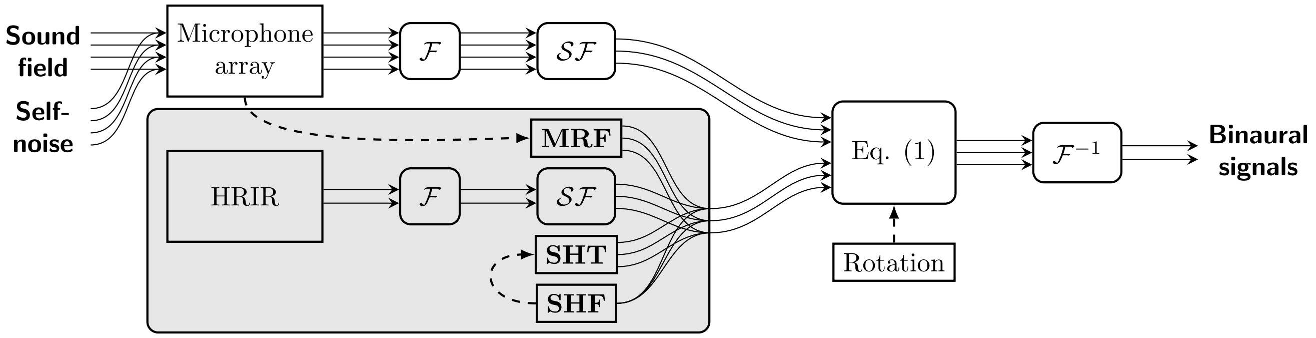

Figure 1: Signal flow graph with pre-computed components during startup (grey); $\mathcal{F}$: Fourier transform, $\mathcal{SF}$: Spherical Fourier transform, **MRF**: modal radial filters, **SHT**: Spherical Harmonics Tapering, **SHF**: Spherical Head Filter.

Latest Developments

Rendering Enhancements

The (binaural) rendering of SMAs by the means of SH expansion faces some inherent limitations due to the effects of *spatial aliasing* and *SH order truncation*. Different strategies have been proposed in the literature to mitigate the perceptual consequences. A diverse set of methods was compared in terms of their perceptual effectiveness recently [10]. The following two approaches have been implemented as part of the `sound_field_analysis-py`[2] toolbox [8], which makes them available in ReTiSAR.

The rendering of decomposed SMA signals with limited SH order leads to the truncation of higher modes in an HRIR data set with high spatial resolution that could otherwise be encoded at higher order. Based on assumptions of a diffuse sound field around a spherical scattering body, the *Spherical Head Filter* (SHF) [2] was proposed. The determined filter compensates for the overall loss of high frequency content in the ear signals as a function of the employed SH rendering order. Since the filter is a global equalization independent of the listener's head orientation, it may be applied at any rendering stage.

Another mitigation of SH order truncation was proposed in the form of *Spherical Harmonics Tapering* (SHT) [9]. The approach employs a modal tapering window depending on the employed rendering order. This leads to a reduction of variations in spatial or coloration properties in the rendered signals during head rotations. Since the tapering window introduces an additional loss of high frequency content, the method can be applied in combination with the previously introduced SHF (slightly adjusted in terms of cut-on frequency).

The aforementioned rendering improvements were incorporated into the ReTiSAR signal flow at the points denoted **SHF** and **SHT** in Fig. 1. Note that HRIR, SHF and MRF constitute filters, while SHT is a frequency independent weighting factor. As indicated in Fig. 1, all static elements are combined after decomposition of the HRIR data set i.e., by element wise multiplication of the complex one-sided spectra in the SH domain. As this is the equivalent operation to convolution in time domain, it requires a sufficiently long buffer to prevent time aliasing. However, since we operate at a designated block length in real-time rendering, we have to make sure that the combination of all elements is confined to the buffer length. This requires assigning of the available FIR coefficients to each of the components HRIR, SHF and MRF. We implemented a basic automatic distribution strategy by means of predefined minimum and maximum limits for the respective parts. Hereby, the filter lengths depend on the chosen rendering configuration i.e., the employed HRIR set, rendering order and SH truncation mitigation. Two exemplary configurations are shown in Fig. 2.

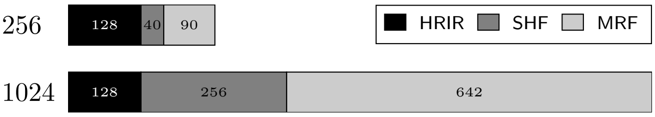

Figure 2: Automatically assigned FIR-filter taps for exemplary rendering configurations at two different block lengths.

ReTiSAR uses the *JACK Audio Connection Kit* (JACK) to act as an interface to the audio hardware and for the internal routing between different rendering stages. We extended the capabilities of the noise-generating JACK client in order to realize arbitrary output volumes for each individual channel. This will be utilized to emulate specific distributions of inter-channel differences in SMA noise (e.g. data on our *Eigenmike* [5]). Also the specific average coloration of the observed self-noise [5] can be replicated in the noise generator by means of an IIR-filter.

Performance Optimizations

In the utilized Python libraries the default word length to store an audio sample i.e., a floating-point number, is 64 bit (*double precision*). Twice the amount of data will be used for the respective complex frequency domain representation (the processing eventually employs one-sided spectra, due to the purely real input signals). Restricting the sample data word length to 32 bit (*single precision*), necessarily leads to a loss in numerical precision and can accumulate arithmetical errors due to rounding. On the other hand, the computational effort to perform *single precision* arithmetic operations is effectively halved.

We adapted every rendering component to operate at *single precision* on request. This is done by enforcing the respective data type when samples are gathered into

[2] `https://github.com/AppliedAcousticsChalmers/sound_field_analysis-py`

Numpy arrays. Upstream, we made sure that the individually optimized pyFFTW Fourier transforms operate under equal constraints. A restriction to *single precision* yielded a performance improvement of a factor of 1.5 to 1.8 (as reported by the JACK system load) for configurations close to the performance limits of the employed computer. A repetition of the instrumental validation of the initial ReTiSAR contribution [7] yielded virtually identical results for either word length. Furthermore, we could not detect any audible differences.

That being said, the JACK infrastructure strictly limits audio time domain streams to *single precision* i.e., intermediate connections between different rendering stages (e.g. ARIR and main binaural renderer) and interaction with the audio interface (e.g. captured microphone and deployed ear signals). It is noteworthy nevertheless that an intermediate utilization of *double precision* accuracy in the SH rendering stage yields no observable benefit for the observed SMA binaural reproduction configurations.

When rendering ARIRs instead of streamed SMA signals, we enabled the option to truncate the impulse responses as a straightforward method of reducing the processing cost if required. ReTiSAR allows to specify a relative cutoff level where all channels decayed by the desired amount below the ARIR's global peak value. Computational savings during the *partitioned overlap-save* convolution for the ARIR rendering are only meaningful in the case that entire blocks can be cut from the calculation. The determined truncation length will be rounded up to an integer multiple of the processing block length consequently.

As an example, the ARIRs of the LBS room in the Cologne data set [14] (reverberation time around 1.8 s) exhibit a length of 33×4096 samples. With an introduced cutoff level of −60 dB the IRs will be truncated to 11×4096 samples, resulting in a reduction of a factor of 3. The JACK system load reports a decrease by a factor of 2.3. Such direct relation between the number of discarded blocks and the reduced JACK system load was also found for other configurations, employing diverse room ARIRs, SMA grids and rendering block lengths.

Convenience Functions

The *Spatially Oriented Format for Acoustics* (SOFA) [11] was developed specifically for the purpose of storing spatial impulse response data. We utilize the pysofaconventions[3] package in order to load HRIR and ARIR data sets of the respective *SimpleFreeFieldHRIR* and *SingleRoomDRIR* conventions. Unfortunately, the conventions do not provide a standardized attribute to store individual *quadrature weights* for the employed spherical sampling grid. The decomposition of SOFA data sets into SH coefficients will therefore always apply the (weighted) least-squares matrix inversion [13].

ReTiSAR supports a wide variety of rendering modes like the auralization of streamed SMA signals (live-captured or from a storage medium) on the one hand, and measured ARIR data sets on the other hand. However, the open-source publication of the code cannot be accompanied by all the required data sets. We therefore implemented a basic infrastructure for dynamically downloading the required files after the renderer has been deployed by the user. Currently, the automatic acquisition is realized in case the data is directly accessible via a provided source URL. The functionality can be easily extended in the future to allow for the unpacking of archives or individual file operations.

The SMA sampling grid (number and distribution of channels) determines the maximum SH order of the sound field that can be extracted stably. We implemented an interface to manually modify the order in the binaural rendering stage during it's runtime. In order to keep the implementation and re-initialization effort during the switching low, the incoming sound field is always decomposed at the initial (maximum) SH order with the non-utilized upper SH modes being discarded. On the other hand, all pre-computed static parts of the running pipeline need to be recreated when switching (cf. Eq. 1 and Fig. 1). All rendering is temporarily paused during the update, which takes a few seconds at maximum.

Furthermore, we implemented an adjustable input delay for the purpose of synchronizing live-captured SMA signals with other simultaneously presented media content like audio or video feeds. A circular delay matrix is added to the input stage of the main binaural renderer in order not to introduce any head-tracking latency.

Conclusions

We extended the capabilities of our Python implementation of a real-time binaural rendering pipeline of spherical microphone array signals. ReTiSAR can now restrict all arithmetic operations to *single precision* yielding considerable performance improvements. The current version is able to render array room impulse response data sets of up to 12th spherical harmonics order on a standard laptop. We implemented a room impulse response truncation as a useful option when rendering array based impulse responses is performed and limited computational resources are at hand.

We made the *Spherical Head Filter* and *Spherical Harmonics Tapering* available to the user, as mitigation strategies for limitations due to *spatial undersampling*. During pre-computation of the static components, the available filters taps are distributed automatically, based on the chosen rendering block size (cf. Fig. 1 and Fig. 2).

ReTiSAR is now able to load ARIR and HRIR data sets in the prevalent *Spatially Oriented Format for Acoustics*. The code publication contains detailed instructions for a wide variety of exemplary rendering configurations, while further contributions are very welcome. A selection of compatible data sets are downloaded automatically by ReTiSAR from their original URLs when invoking the example configurations that we provide.

Acknowledgement

We thank Facebook Reality Labs for endorsing the publication of the source code and financing this project.

[3] https://github.com/andresperezlopez/pysofaconventions

References

[1] Amir Avni, Jens Ahrens, Matthias Geier, Sascha Spors, Hagen Wierstorf, and Boaz Rafaely. Spatial perception of sound fields recorded by spherical microphone arrays with varying spatial resolution. *Journal of the Acoustical Society of America*, 133(5):2711–2721, 2013.

[2] Zamir Ben-Hur, Fabian Brinkmann, Jonathan Sheaffer, Stefan Weinzierl, and Boaz Rafaely. Spectral equalization in binaural signals represented by order-truncated spherical harmonics. *Journal of the Acoustical Society of America*, 141(6):4087–4096, 2017.

[3] Benjamin Bernschütz. *Microphone Arrays and Sound Field Decomposition for Dynamic Binaural Recording.* Phd thesis, Technische Universität Berlin, 2016.

[4] Benjamin Bernschütz, Christoph Pörschmann, Sascha Spors, and Stefan Weinzierl. SOFiA Sound Field Analysis Toolbox. In *International Conference on Spatial Audio*, pages 7–15, Detmold, Germany, 2011. Verband Deutscher Tonmeister e.V.

[5] Hannes Helmholz, Jens Ahrens, David Lou Alon, Sebastià V. Amengual Garí, and Ravish Mehra. Evaluation of Sensor Self-Noise in Binaural Rendering of Spherical Microphone Array Signals. In *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5, Barcelona, Spain, 2020. IEEE.

[6] Hannes Helmholz, David Lou Alon, Sebastià V. Amengual Garí, and Jens Ahrens. Instrumental Evaluation of Sensor Self-Noise in Binaural Rendering of Spherical Microphone Array Signals. In *Forum Acusticum*, pages 1–8, Lyon, France, 2020. EAA.

[7] Hannes Helmholz, Carl Andersson, and Jens Ahrens. Real-Time Implementation of Binaural Rendering of High-Order Spherical Microphone Array Signals. In *Fortschritte der Akustik – DAGA 2019*, pages 1462–1465, Rostock, Germany, 2019. Deutsche Gesellschaft für Akustik.

[8] Christoph Hohnerlein and Jens Ahrens. Spherical Microphone Array Processing in Python with the sound_field_analysis-py Toolbox. In *Fortschritte der Akustik – DAGA 2017*, pages 1033–1036, Kiel, Germany, 2017. Deutsche Gesellschaft für Akustik.

[9] Christoph Hold, Hannes Gamper, Ville Pulkki, Nikunj Raghuvanshi, and Ivan J. Tashev. Improving Binaural Ambisonics Decoding by Spherical Harmonics Domain Tapering and Coloration Compensation. In *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 261–265, Brighton, UK, 2019. IEEE.

[10] Tim Lübeck, Hannes Helmholz, Johannes M. Arend, Christoph Pörschmann, and Jens Ahrens. Perceptual Evaluation of Mitigation Approaches of Impairments due to Spatial Undersampling in Binaural Rendering of Spherical Microphone Array Data. *Journal of the Audio Engineering Society*, pages 1–12 (submitted).

[11] Piotr Majdak, Yukio Iwaya, Thibaut Carpentier, Rozenn Nicol, Matthieu Parmentier, Agnieszka Roginska, Yôiti Suzuki, Kanji Watanabe, Hagen Wierstorf, Harald Ziegelwanger, and Markus Noisternig. Spatially Oriented Format for Acoustics: A Data Exchange Format Representing Head-Related Transfer Functions. In *AES Convention 134*, pages 262–272, Rome, 2013. Audio Engineering Society.

[12] Munhum Park and Boaz Rafaely. Sound-field analysis by plane-wave decomposition using spherical microphone array. *Journal of the Acoustical Society of America*, 118(5):3094–3103, 2005.

[13] Nico Sneeuw. Global spherical harmonic analysis by least-squares and numerical quadrature methods in historical perspective. *Geophysical Journal International*, 118(3):707–716, 1994.

[14] Philipp Stade, Benjamin Bernschütz, and Maximilian Rühl. A Spatial Audio Impulse Response Compilation Captured at the WDR Broadcast Studios. In *27th Tonmeistertagung – VDT International Convention*, pages 551–567, Cologne, Germany, 2012. Verband Deutscher Tonmeister e.V.