Acta Acustica united with Acustica
Vol. 105 (2019) 566 – 578

DOI 10.3813/AAA.919337

A Toolbox for Rendering Virtual Acoustic Environments in the Context of Audiology

Giso Grimm[1,2], Joanna Luberadzka[1], Volker Hohmann[1,2]

[1] Medizinische Physik and Cluster of Excellence "Hearing4all", Department of Medical Physics and Acoustics, University of Oldenburg, Germany. g.grimm@uni-oldenburg.de

[2] HörTech gGmbH, Marie-Curie-Str. 2, 26129 Oldenburg, Germany

Summary

A toolbox for creation and rendering of dynamic virtual acoustic environments (TASCAR) that allows direct user interaction was developed for application in hearing aid research and audiology. This technical paper describes the general software structure and the time-domain simulation methods, i.e., transmission model, image source model, and render formats, used to produce virtual acoustic environments with moving objects. Implementation-specific properties are described, and the computational performance of the system was measured as a function of simulation complexity. Results show that on commercially available commonly used hardware the simulation of several hundred virtual sound sources is possible in the time domain[a].

PACS no. 43.55.Ka, 43.60.Dh

[a] Parts of this study have been presented at the Linux Audio Conference, Mainz, Germany, 2015.

1. Introduction

Hearing aids have been evolving from simple amplifiers to complex signal processing devices. Current hearing devices typically contain spatially sensitive algorithms, e.g., directional microphones, direction of arrival estimators, or binaural noise reduction, as well as automatic classification of the acoustic environment that is used for context-adaptive processing and amplification [1]. Several of these features cannot be tested in the current lab-based setups for hearing-aid evaluation, because they employ rather simple acoustic configurations. For example, direction of arrival estimation might work well in static settings with few sound sources and low reverberation, but may fail in the field where the algorithm is exposed to head movements, moving sources, diffuse background noise and reverberation. Furthermore, it was shown in several studies that hearing aid performance depends on the spatial complexity of the environment, and that the hearing aid performance in simple laboratory conditions is not a good predictor of the performance in more realistic environments or in the real life [2, 3, 4, 5, 6]. Finally, recent developments in hearing aid technology led to an increased level of interaction between the user, the environment and the hearing devices, e.g., by means of motion interaction [7, 8], gaze direction [9] or even with brain-computer interfaces [10]. The highest level of ecological validity is achieved in field tests. Due to the rather low level of control, however, reproducibility is not given, and sensitivity is low for most field test methods. Established laboratory tests are highly reproducible and sensitive, but lack ecological validity. Thus, for an improved assessment of hearing aid benefit as well as for the development and evaluation of user interaction techniques for hearing devices, a reproduction system with scalable complexity from static settings with only a few sources up to complex dynamic and more realistic listening environments in the laboratory are required. Such systems are not seen as a replacement of neither simple laboratory tests nor field studies, but rather as an option to systematically fill the gap between the laboratory with high control and the field tests with highest ecological validity.

Advances in computer technology in combination with recent multi-channel reproduction [11, 12, 13] and acoustic simulation methods [14, 15, 16] allow for the reproduction of virtual acoustic environments in the laboratory. Limitations in reproduction and simulation quality have been studied in terms of perceptual effects [17, 18] as well as in terms of technical accuracy of hearing aid benefit prediction [19, 20]. These studies support the general applicability of virtual acoustic environments to hearing-aid evaluation and audiology, but show that care must be taken in designing the simulation and reproduction methods, to ensure that the outcome measures are not biased by the artifacts of the applied methods.

Received 15 August 2018,
accepted 3 April 2019.

Several further requirements apply when using virtual acoustic environments in hearing research and audiology. To allow for a systematic evaluation of hearing device performance, virtual acoustic environments need to be reproducible and scalable in their complexity. Early reflections and late reverberation affect speech intelligibility, localization performance, distance perception and movement detection in normal-hearing and hearing-impaired listeners. Therefore, the benefit provided by hearing devices with spatial filtering, e.g., beamforming or de-reverberation, is also affected by these factors [2], and thus it is essential for more realistic hearing device evaluation to simulate all of these features. For assessment of user interaction, but also for the analysis of hearing aid benefit, simulation of the effects of motion of listeners and sources is required. These effects do not only include time-variant spatial cues, but also Doppler-shift and time-variant spectral cues due to comb filtering. Interactivity is seen in the sense that the audio content and the motion or position of sources can be controlled in real-time, e.g., by adaptive measurement procedures, or for simulation of self-motion. An example for interactivity is an experiment with adaptive movement detection [21]. Another application of interactivity are situations in which self-motion is applied to the simulation, in order to achieve acoustically correct stimuli, e.g., when the listener is close to a virtual reflector.

The proposed simulation tool implements a system similar to the DIVA system [22]. Most other existing virtual acoustic environment engines target authentic simulations for room acoustics [23, 24], resulting in a large computational complexity. They typically render impulse responses for off-line analysis or auralization and thus do not allow studying motion and user interaction. Furthermore, the output is typically used for human perception experiments only, and may not work for analysis with multi-channel microphone arrays, e.g., beam-forming techniques in hearing devices. To overcome this problem, the tool MCRoomSim was developed [25], for simulation of shoe-box shaped rooms. In a study with blind subjects [26], a real-time interpolation of pre-rendered listening positions was used, based on render outputs of complex room acoustic tools. Other interactive tools, e.g., the SoundScapeRenderer [27], do not provide all features required here, such as room simulation and diffuse sound field handling. The tool RAVEN [28] provides a combination of room acoustic modeling with dynamic interaction, by providing a time-variant simulation of the direct path, fast updates of early reflections and slower updates of late reverberation. The approach of the tool EVERTims [29] is similar to that of RAVEN. A ray-tracing algorithm is used to generate impulse responses. These impulse responses are updated upon movement, and rendered in higher order Ambisonics. However, these simulation tools require higher computational performance and at time of writing do not provide a time domain simulation of early reflections. Thus, they can not render continuous time-variant comb filter effects in case of moving sources or receivers in the presence of reflecting surfaces, e.g., as required for plausible simulation of moving cars in outdoor simulations. The LoRA system [30] renders virtual acoustics as created by room acoustic simulation tools via higher order Ambisonics and multi-channel loudspeaker reproduction, for auralization with hearing impaired subjects. This way, head rotation and within a limited area, also translational self-motion is possible in the virtual acoustic environment. A similar approach is used in SOFE [31]. Here, latency constraints do not allow for low-delay interaction.

To accommodate the requirements listed above, a toolbox for acoustic scene creation and rendering (TASCAR) was developed as an open source tool [32, 33] with commercial support [34]. The aim of the toolbox is to interactively render complex and time-varying virtual acoustic environments via loudspeakers or headphones. For a seamless integration into existing measurement tools of psycho-acoustics and audiology, low-delay real-time processing of external audio streams in the time domain is applied, and interactive modification of the geometry is possible. Virtual environments in TASCAR are edited in human-readable text format (XML). Paths and the shapes of objects, e.g., reflectors, can be imported from the open 3D-modeling tool "blender" [35]. The current virtual environment can be displayed in a 2-dimensional projection. Positions of all primary sources and image sources can be exported graphically and as a table. For interaction with sensors and tools, and embedding into laboratory environments, e.g., for data logging, the open sound control (OSC) protocol [36] and the lab streaming layer (LSL) [37] are used.

The proposed simulation tool is based on established simulation methods, e.g., geometric image source model, and render formats, such as VBAP [12] or HOA [13]. The physical and perceptual properties of these render methods have been extensively studied [38, 39, 40, 41, 27, 42, 43, 17]. The limitations for applications in hearing aid evaluation differ from perceptual limitations [19]. They depend on the sensitivity of hearing aid algorithms and the applied hearing aid performance measures on spatial aliasing artifacts of the render methods. Thus the optimal render method depends on the context of a specific application of the proposed simulation tool. Based on the data by [19], a specific scene can be designed such that it meets the requirements of an application-specific receiver, e.g., a human head with two-microphone hearing aids on each ear.

An overview over a number of possible applications is shown in Figure 1. The simplest application of TASCAR is to play back a pre-defined virtual acoustic environment via multiple loudspeakers (Figure 1.a). For subjective audiological or psycho-acoustic measurements in virtual acoustic environments, without hearing aids or aided with conventional hearing aids, the audio input of virtual sound sources can be provided by external measurement tools (Figure 1.b). The toolbox can also be applied to assess hearing aid (HA) performance in virtual acoustic environments, based on instrumental measures, or with human listeners, e.g., in combination with a Master Hearing Aid

[44, 45]. Subjective or instrumental evaluation of research hearing aids can be performed by feeding the output of the virtual acoustic environment directly to the inputs of a research hearing aid (Figure 1.c), similar as proposed by [46]. An example study of this use case can be found in [6], where hearing aid performance in eight different virtual acoustic environments of different spatial complexity was assessed. Test stimuli as well as the configuration of virtual acoustic environment and the research hearing aid can be controlled from the measurement platform, e.g., MATLAB or GNU/Octave (Figure 1.d). Motion data can also be recorded from motion sensors or controllers, to interact with the environment in real-time, or for data logging (Figure 1.e).

These use cases serve as an illustration of typical applications of TASCAR. The interfaces of the toolbox allow for a large number of applications. The strength of the proposed toolbox is its design towards reproducible scientific applications: the implementation of the data logging ensures a documentation of version numbers, allows for integration of many data streams in a central data logging with a common time-line, and documents the session file used for the recording of data. The xml-based scene definition format allows for detailed and flexible way of describing the virtual scene. All of these features were implemented to overcome the otherwise often experienced problem, that after the publication of data it is almost impossible to reproduce the results due to undocumented properties of the applied simulation components (virtual acoustic modelling or simulated hearing aid signal processing).

This technical paper aims at providing a technical reference of the proposed tool by describing the underlying simulation and rendering methods, and their specific implementation. Furthermore, the structure of the toolbox in applications of hearing aid evaluation and audiology is explained. A measurement of the computational performance and its underlying factors is provided to allow for an estimation of maximum simulation complexity in relation to the available computing power. This paper also serves as a technical reference for the TASCAR open source software (TASCAR/GPL). A list of environments rendered with the proposed tool can be found in [47].

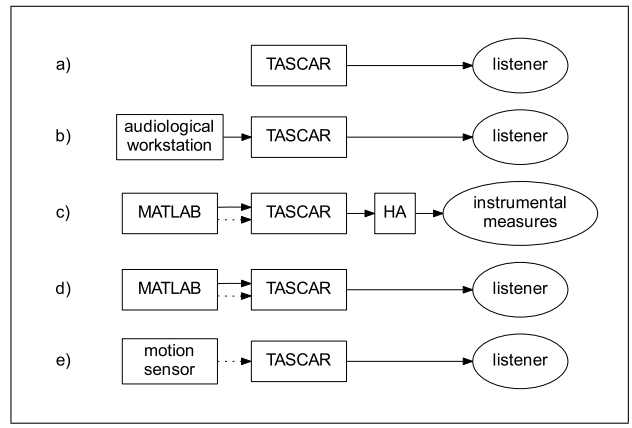

Figure 1. Example applications of TASCAR and its interaction. Solid arrows indicate audio signals, dashed arrows represent control information, e.g., geometry data.

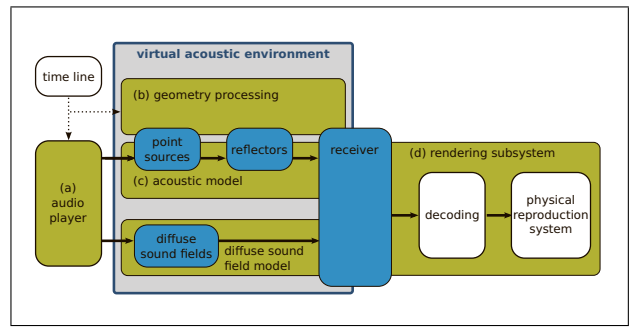

Figure 2. The major components of TASCAR are the audio player (a), the geometry processor (b), the acoustic model (c) and the rendering subsystem (d). Point sources and diffuse sound fields are the interface between the audio player and the acoustic model. Receivers are the interface between the acoustic model and the rendering subsystem.

General structure

The structure of the proposed toolbox can be divided into four major components (see Figure 2 for an overview): The audio player (block a in Figure 2) serves as a source of audio signals. The geometry processor (block b) controls position and orientation of objects over time. The acoustic model (blocks c) simulates sound propagation, room acoustics and diffuse sound fields. Finally, the rendering subsystem (block d) renders the output of the acoustic model for a physical reproduction system.

A virtual acoustic environment in TASCAR is called *scene* and defines a space containing several types of objects: point sources (e.g., speakers, distinct noise sources), diffuse sound fields (e.g., remote traffic, babble noise), receivers (e.g., dummy head), reflectors (e.g., boundaries of a room) and obstacles. Audio content is delivered to source objects either by the internal audio player module, or externally e.g., from physical sources, audiological measurement tools, or digital audio workstations (DAW).

Objects in a scene can change their positions and orientations over time. Information about the object geometry at a given time is taken either from sampled trajectories, from algorithmic trajectory generators, or from external devices, such as head tracking systems or game controllers. Geometry information is exploited in the acoustic model to modify the input audio signals delivered by the audio player. Modifications performed by the acoustic model mimic basic acoustic properties like distance law, reflections and air absorption. Geometry data can also be exchanged with external modules, e.g., game engines, to make the visualization consistent with the acoustic scene content.

At the final stage of the acoustic model, there is a receiver model, which encodes the modified signals into a receiver type specific render format, used subsequently by the rendering subsystem for the reproduction of the simulated environment on a physical reproduction system. The

resulting sound corresponds to the time-variant spatial arrangement of the objects in the virtual scene.

2. Simulation methods

2.1. Geometry processing

Each object in a scene is determined by its position $\mathbf{p}(t)$ and orientation $\mathbf{\Omega}(t)$ in space at a given time t. Position is defined in Cartesian coordinates $\mathbf{p} = (p_x, p_y, p_z)$, and orientation is defined in the Euler angles, $\mathbf{\Omega} = (\Omega_z, \Omega_y, \Omega_x)$, where Ω_z is the rotation around the z-axis, Ω_y around the y-axis and Ω_x around the x-axis.

Trajectories $\mathbf{\Gamma}$ for a moving object are created by specifying the position and orientation for more then one point in time:

$$\begin{aligned}\mathbf{\Gamma_p} &= \left\{\mathbf{p}(t_1), \mathbf{p}(t_2), \mathbf{p}(t_3), \dots\right\}\\ \mathbf{\Gamma_\Omega} &= \left\{\mathbf{\Omega}(t_1), \mathbf{\Omega}(t_2), \mathbf{\Omega}(t_3), \dots\right\},\end{aligned}$$

where $t_1, t_2, \dots \in \mathbb{R}$ are the sample times of the trajectory. The time-variant position $\mathbf{p}(t)$ is linearly interpolated between sample times of $\mathbf{\Gamma_p}$, either in Cartesian coordinates, or in spherical coordinates relative to the origin, respectively. The time-variant orientation $\mathbf{\Omega}(t)$ is linearly interpolated from $\mathbf{\Gamma}_\Omega$, in Euler angles. To apply the orientation to objects, a rotation matrix $\mathbf{O}$ is calculated from the Euler angles:

$$\mathbf{O} = \mathbf{O}_x\mathbf{O}_y\mathbf{O}_z, \tag{1}$$

with

$$\mathbf{O}_x = \begin{pmatrix}1 & 0 & 0\\ 0 & \cos(\Omega_x) & -\sin(\Omega_x)\\ 0 & \sin(\Omega_x) & \cos(\Omega_x)\end{pmatrix},$$

$$\mathbf{O}_y = \begin{pmatrix}\cos(\Omega_y) & 0 & \sin(\Omega_y)\\ 0 & 1 & 0\\ -\sin(\Omega_y) & 0 & \cos(\Omega_y)\end{pmatrix} \text{ and}$$

$$\mathbf{O}_z = \begin{pmatrix}\cos(\Omega_z) & -\sin(\Omega_z) & 0\\ \sin(\Omega_z) & \cos(\Omega_z) & 0\\ 0 & 0 & 1\end{pmatrix}.$$

The recommended spatial-temporal sampling depends on the object type and application: the orientation of receivers should have a sufficiently high sampling to avoid orientation jumps larger than the spatial resolution of the reproduction system. Trajectories of sound sources with time-variant velocity require sufficient sampling for a smooth pitch or coloration perception due to the Doppler shift.

Although the different types of objects are not directly related to each other in terms of acoustic modeling, their geometry is treated in the same way. As an example, dynamic trajectories of receivers can be used to simulate self-motion, moving source objects can be used to simulate moving sounds such as cars, and dynamic obstacles and reflectors can be used to simulate incoming trains in a train station.

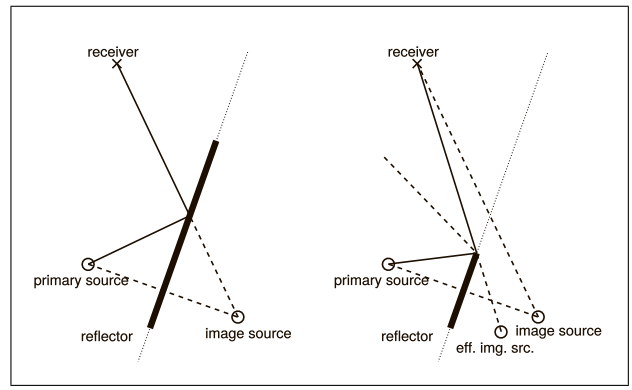

Figure 3. Schematic sketch of the image model geometry. Left sketch: 'specular' reflection, i.e., the image source is visible within the reflector; right sketch: 'edge' reflection.

2.2. Acoustic model

For each sound source object k, the acoustic model modifies its associated original source signal $x(t)$ delivered by the audio player using geometry data into an output signal $y(t)$ that is then used as input signal to a receiver. The performed computations simulate basic acoustic phenomena as described below. Signals $y(t)$ serve at the subsequent stage to calculate the output of a receiver (see Section on render formats below).

The acoustic model consists of the source model (omnidirectional or frequency-dependent directivity), the transmission model simulating sound propagation, an image source model (ISM), which depends on the reflection properties of the reflecting surfaces as well as on the 'visibility' of the reflected image source, and a receiver model, which encodes the direction of the sound source relative to the receiver into a receiver-type specific render format.

2.2.1. Image source model

Early reflections are generated with a geometric ISM, i.e., reflections are simulated for each reflecting plane surface with polygon-shaped boundary by placing an image source at it's apparent position behind the reflector. Each image source is rendered in the time domain, in the same way as primary sources. This is different to the more efficient "shoe-box" ISMs commonly used in room acoustic simulations [14], which calculate impulse responses by solving the wave equations. For a first order ISM, each pair of primary source and reflector face creates an image source, where the plane on which the reflector lies is a symmetry axis between the primary and image source (see Figure 3). The image source position $\mathbf{p}_{img}$ is determined by the closest point on the (infinite) reflector plane $\mathbf{p}_{\text{cut}}$ to the source $\mathbf{p}_{\text{src}}$: $\mathbf{p}_{img} = 2\mathbf{p}_{\text{cut}} - \mathbf{p}_{\text{src}}$.

For higher order ISMs, lower order image sources are treated as primary sources leading to higher order image sources.

The image source position itself is independent of the receiver position. However, for finite reflectors there are two types of reflections in TASCAR, and depending on the receiver position it is determined which reflection type

is executed (see Figure 3). If the intersection point $\mathbf{p}_{is}$ of the line connecting the image source with the receiver and the reflector plane lies within the reflector boundaries, the image source is 'visible' in the reflector, and a 'specular' reflection is applied. If $\mathbf{p}_{is}$ is not within the reflector boundaries, the source is 'invisible' from a receiver perspective and the 'edge reflection' is applied. For 'edge' reflections, the image source position is adjusted so that the distance between the source and receiver remains unchanged, whereas the receiver, edge of the reflector and the apparent source position form one line (see Figure 3, right panel). The angle θ by which the image source is shifted to create adjusted image source controls a soft-fade gain by which the source signal is multiplied g:

$$g = \cos(\theta)^{\kappa} \tag{2}$$

The coefficient $\kappa = 2.7$ was chosen for a rough approximation of diffraction of speech-shaped signals and medium-sized reflectors. For a more elaborated diffraction model see, e.g., [48, 49]. If a receiver or a sound source are behind the reflector, the image source is not rendered. A reflector object has only one reflecting side in the direction of the face normal.

To simulate the reflection properties of a reflector object, the source signal is filtered with a first order low pass filter determined by a reflectivity coefficient ρ, and a damping coefficient δ, which can be specified for each reflector object:

$$y(t) = \delta y(t - f_s^{-1}) + \rho x(t). \tag{3}$$

Higher order reflections are achieved by applying this filter for each reflection. In room acoustics material properties are commonly defined by frequency dependent absorption coefficients $\alpha(f)$. These can be calculated from the reflection filter coefficients ρ and δ by

$$\alpha(f) = \left(1 - \left|\rho \frac{1-\delta}{1-\delta \mathrm{e}^{-\mathrm{i}2\pi f f_s^{-1}}}\right|\right)^2. \tag{4}$$

The filter coefficients ρ and δ can be derived from frequency dependent absorption coefficients by minimization of the mean-square error between desired $\tilde{\alpha}(f)$ and $\alpha(f)$ derived from the filter coefficients.

2.2.2. Source directivity

For the simulation of primary source directivity, the receiver position relative to the source

$$\mathbf{p}_{\mathrm{rec,rel}} = \mathbf{O}_{\mathrm{src}}^{-1}(\mathbf{p}_{\mathrm{rec}} - \mathbf{p}_{\mathrm{src}}) \tag{5}$$

is calculated. At time of writing, source directivity in TASCAR is supported only for primary sources due to the numerical complexity of the geometry update: modelling source directivity requires the update of orientation at each reflection, which roughly doubles the computational complexity of the geometry update. For simulations in the context of audiology this is not a critical limitation as long as extreme spatial configurations, e.g., source is facing to the reflector so that the image source has a higher amplitude at the receiver than the primary source, are avoided. Frequency-dependent directivity with omnidirectional characteristics at low frequencies and higher directivity at high frequencies is achieved by controlling a low-pass filter by the angular distance between the receiver and the source direction. The normalized relative receiver position $\tilde{\mathbf{p}}_{\mathrm{rec,rel}}$ is

$$\tilde{\mathbf{p}}_{\mathrm{rec,rel}} = \frac{\mathbf{p}_{\mathrm{rec,rel}}}{||\mathbf{p}_{\mathrm{rec,rel}}||} \tag{6}$$

The cosine of the angular distance is then $\tilde{p}_{x,\mathrm{rec,rel}}$. The cut-off frequency f_{6dB} defines the frequency, for which -6 dB at ± 90 degrees are achieved. With $\xi = \pi f_{6\mathrm{dB}}/f_s/\log(2)$, a first order low-pass filter with the recursive filter coefficient c_{lp},

$$c_{lp} = \left(\frac{1}{2} - \frac{1}{2}\tilde{p}_{x,\mathrm{rec,rel}}\right)^{\xi(f_{\mathrm{cut}})}, \tag{7}$$

is applied to the signal, to achieve the frequency-dependent directivity, or in other words, the direction-dependent frequency characteristics.

2.2.3. Transmission model

The transmission model simulates the delay, attenuation and air absorption, which depend on the distance $r(t)$ between the sound source (primary or image source) and the receiver, as well as attenuation, caused by obstacles between source and receiver. Point sources follow a $1/r$ sound pressure law, i.e., doubling the distance r results in half of the sound pressure. Air absorption is approximated by a simple first order low-pass filter model with the filter coefficient a_1 controlled by the distance:

$$a_1 = \mathrm{e}^{-\frac{r(t) f_s}{c\alpha}}, \tag{8}$$

where f_s is the sampling frequency and c the speed of sound. The empiric constant $\alpha = 7782$ was manually adjusted to provide appropriate values for distances below 50 meters. This approach is very similar to that of [50] who used an FIR filter to model the frequency response at certain distances. However, in this approach the distance parameter r can be varied dynamically. The distance dependent part of the transmission model without obstacles can then be written as

$$y(t) = a_1 y(t - f_s^{-1}) + \left(1 - a_1\right)\frac{x(t - r(t)c^{-1})}{r(t)}, \tag{9}$$

where $x(t)$ is the source signal at time t, and $y(t)$ is the output audio signal of the transmission model. The time-variant delay line uses either nearest neighbor interpolation or sinc interpolation, depending on the user needs and computational performance of the computing system.

Obstacles are modeled by plane surfaces with polygon-shaped boundaries. The acoustic signal is split into a direct path, which is attenuated by the obstacle-specific frequency-independent attenuation a_o, and an indirect

path, to which a simple diffraction model is applied. The diffracted path is filtered with a second order low pass filter which is controlled by the shortest path from the source via the obstacle boundary to the receiver. With the angle θ_o between the connection from the intersection point of the shortest path with the obstacle boundary to the source position, and the connection from the receiver position to the intersection point, the cut-off frequency of the low-pass filter is

$$f_o = 3.8317 \frac{c}{2\pi a \sin(\theta_o)}, \tag{10}$$

with the aperture $a = 2\sqrt{A/\pi}$ defined as the diameter of a circle with the same area A as the obstacle polygon. This simple diffraction model is based on the diffraction on the boundary of a circular disc [51], however, position-dependent notches are not simulated. The diffracted signal is weighted with $1 - a_o$ and added to the attenuated signal. Again, for a more elaborated diffraction model, see e.g., [48, 49].

2.2.4. Diffuse sound fields

Diffuse sound fields, e.g., diffuse background noise or diffuse reverberation rendered in external tools [16], or recordings of environmental sounds with low spatial resolution, are added in first order Ambisonics (FOA) format. No distance law is applied here; instead, diffuse sound fields have a rectangular spatial range box, i.e., they are only rendered if the receiver is within their range box, with a von-Hann ramp at the boundaries of the range box. These range boxes can be used to achieve interactive simulations with position-dependent diffuse sound fields, e.g., in a simulation of a street environment to allow for transitions between traffic noise and more natural sounds while departing from a busy street. In simulation of closed spaces the range box offers a simple method to allow transitions between rooms with different diffuse reverberation. In this case, the size of the range box is typically adjusted to match the dimension of the simulated rooms. Position and orientation of the range box can vary with time. The diffuse field is rotated by the difference between box orientation and receiver orientation. Direct reproduction of FOA signals leads to a low diffuseness and introduces coloration artefacts caused by self-motion within the reproduction system. To compensate these problems, the rendered signals are decorrelated; see below for a detailed description.

Diffuse reverberation is not simulated in TASCAR. To use diffuse reverberation, the input signals of the ISM can be passed to external tools which return FOA signals, e.g., feedback-delay networks or convolution with room impulse responses in FOA format [52]. A smooth transition between early reflections from the ISM and diffuse reverberation based on room impulse responses can be achieved by removing the first reflections from the impulse responses. To account for position-independent late reverberation, room receivers can render independent from the distance between source and receiver, e.g., the transmission model can be replaced by a room-volume dependent fixed gain. An integration of the FDN-based diffuse reverberation method RAZR [16] into the proposed toolbox is planned for the future.

Distance perception in human listeners is believed to be dominated by the direct-to-reverberant ratio [53]. In the proposed simulation tool with a simple ISM and position-independent externally generated late reverberation, the distance perception may depend on simulation parameters. Thus, in a previous study the distance perception and modeling with room-acoustic parameters in simulations with TASCAR was evaluated [18]. It was shown in a comparison of binaural recordings in a real room and a simulation of the same geometry that in the simulation a distance perception similar to real rooms can be achieved.

2.2.5. Receiver model

The interface between the acoustic model and the rendering subsystem is the receiver. A receiver is a virtual object in a scene that captures the sound at its position in the scene. Receivers can be realized by different types. Each receiver type generates output signals in a specific render format. Directivity and number of output channels depend on the render format and its configuration, e.g., a virtual microphone returns a single channel, and the directivity can be configured to be omni-directional, cardioid or figure-of-eight. This means that the render format determines the number of channels and the method of encoding the relative spatial information into a multi-channel audio signal. Signals from the transmission models belonging to all sound sources are summed up after direction-dependent processing. The output signal of a receiver is

$$\mathbf{z}(t) = \left(z_1(t), z_2(t), \ldots, z_N(t)\right). \tag{11}$$

The receiver functionality can be split into the *panning* or directional encoding of primary and image sources $y(t)$, and the *decoding* of diffuse sound field signals $\mathbf{f}(t)$ in first order Ambisonics format with Furse-Malham normalization ('B-format'):

$$\mathbf{z}(t) = \underbrace{\sum_{k=1}^{K} \mathbf{w}(\mathbf{p}_{rel,k}) y_k(t)}_{\text{panning}} + \underbrace{\mathbf{H}_d \sum_{l=1}^{L} \mathbf{D}\hat{\mathbf{O}}_{\text{rec}}^{-1} \mathbf{f}_l(t)^T}_{\text{diffuse decoding}}. \tag{12}$$

In the panning part, the driving weights $\mathbf{w} = (w_1, w_2, \ldots, w_N)$ depend on the direction of the relative source position in the receiver coordinate system, $\mathbf{p}_{rel,k} = \mathbf{O}_{\text{rec}}^{-1}(\mathbf{p}_k - \mathbf{p}_{\text{rec}})$; $\mathbf{O}_{\text{rec}}$ is the receiver orientation matrix, and $\mathbf{p}_k$ is the position of the k-th sound source. $y_k(t)$ is the output signal of the transmission model, i.e., it contains the distance-dependent gain, air absorption and obstacle attenuation, for the k-th source; K is the number of all primary and image point sources.

In the diffuse decoding part, $\mathbf{D}$ is the render format specific first order Ambisonics decoding matrix for the w, x, y and z channels of the first order Ambisonics signal,

$$\mathbf{D} = \begin{pmatrix} d_{1,w} & d_{1,x} & d_{1,y} & d_{1,z} \\ \vdots & \vdots & \vdots & \vdots \\ d_{n,w} & d_{n,x} & d_{n,y} & d_{n,z} \end{pmatrix}, \tag{13}$$

and $\hat{\mathbf{O}}_{\text{rec}}^{-1}$is the rotation matrix for first order Ambisonics signals, to compensate the receiver orientation. $\mathbf{f}_l$ is the first order Ambisonics signal of the l-th diffuse sound field, rotated by the sound field orientation; L is the number of all diffuse sound fields, including diffuse reverberation inputs. For all loudspeaker-based render methods (e.g., nearest speaker or VBAP, see below), the FOA signal is decoded to all loudspeakers. To achieve a high diffuseness and to reduce artefacts caused by self-motion, all-pass filters of 50 ms length and random phase response, $\mathbf{H}_d$, are applied to the decoded FOA signals. This method corresponds to the diffuse sound rendering as described by [54]. As an alternative in case of two-dimensional HOA rendering, diffuse upsampling similar to [55] can be applied: the FOA signal at the receiver, $\hat{\mathbf{O}}_{\text{rec}}^{-1}\mathbf{f}(t)$, is split into two spectrally disjoint signals using recursive comb-filters. Given a maximum Ambisonics order o, the coefficients of $\mathbf{D}$ for orders 2 to o correspond to a rotation by $\pm\alpha$ for the two signals. The rotation angle α is typically set to 45 degrees.

2.3. Render formats

The render formats of TASCAR can be divided into three categories: *Virtual microphones* simulate the characteristics of microphones. They primarily serve as a sensor in a scene. *Loudspeaker-based* render formats create signals which can drive real or virtual loudspeakers, used for auralization of virtual scenes. *Ambisonics* render formats reproduce the scenes to first, second or third order Ambisonics format, which can be further processed using external decoders or other Ambisonics tools. Receivers can render either for three-dimensional reproduction or for two-dimensional reproduction. In both cases, the directional information of the relative source position is encoded in the normalized relative source position,

$$\tilde{\mathbf{p}}_{rel} = \frac{\mathbf{p}_{rel}}{||\mathbf{p}_{rel}||}. \tag{14}$$

However, in the two-dimensional case $\mathbf{p}_{rel}$ is projected onto x, y-plane before the normalization by setting its z-component to zero. The acoustic model, containing all distance-dependent effects, and the ISM, are calculated based on the three-dimensional relative source position.

2.3.1. Virtual microphones

The virtual microphone render format has a single output channel. The driving weight is

$$w = 1 + a(\tilde{p}_{rel,x} - 1). \tag{15}$$

It's directivity pattern can be controlled between omni-directional and figure-of-eight with the directivity coefficient a; with $a = 0$ this is an omni-directional microphone, with $a = \frac{1}{2}$ a standard cardioid, and with $a = 1$ a figure-of-eight. The diffuse decoding matrix is

$$\mathbf{D} = \begin{pmatrix} \sqrt{2}(1-a) & a & 0 & 0 \end{pmatrix}. \tag{16}$$

The factor $\sqrt{2}$ of the w-channel is needed to account for the Furse-Malham normalization of the diffuse signals.

2.3.2. Loudspeaker-based render formats

This class of render formats contains all types which render the signals directly to a loudspeaker array. The number N and position $\mathbf{s}_n$ of loudspeakers can be user-defined; $\tilde{\mathbf{s}}_n$ is the normalized speaker position. A measure of angular distance between a source and a loudspeaker is $d_n = 1 - \tilde{\mathbf{s}}_n\tilde{\mathbf{p}}_{rel}^T$. The most basic speaker-based render format is *nearest speaker panning* (NSP). The driving weights are:

$$w_n = \begin{cases} 1 & n = \arg\min\{d_n\}, \\ 0 & \text{otherwise.} \end{cases} \tag{17}$$

NSP has the advantage of no coloration artefacts compared to methods like HOA or VBAP [56]. However, for the reproduction of moving sources it is not sufficient. Another commonly used speaker-based render format is *vector-base amplitude panning* (VBAP) [12]. In the 2-dimensional case, the two loudspeakers n_1 and n_2 which are closest to the source are chosen. A gain vector $(g_{n_1}, g_{n_2})^T$ based on the normalized speaker positions and the normalized relative source position in the x, y-plane is defined:

$$\begin{pmatrix} g_{n_1} \\ g_{n_2} \end{pmatrix} = \begin{pmatrix} \tilde{s}_{n_1,x} & \tilde{s}_{n_2,x} \\ \tilde{s}_{n_1,y} & \tilde{s}_{n_2,y} \end{pmatrix}^{-1} \begin{pmatrix} \tilde{p}_{\text{rel},x} \\ \tilde{p}_{\text{rel},y} \end{pmatrix} \tag{18}$$

Then the driving weights are

$$\begin{pmatrix} w_{n_1} \\ w_{n_2} \end{pmatrix} = \frac{1}{\sqrt{g_{n_1}^2 + g_{n_2}^2}} \begin{pmatrix} g_{n_1} \\ g_{n_2} \end{pmatrix}. \tag{19}$$

VBAP with 3-dimensional reproduction selects the triangle from the convex hull formed by the loudspeaker arrangement, which is intersected by the connection line between receiver and virtual sound source. For ambisonic panning with arbitrary order, the signal of each source is encoded into horizontal Ambisonics format (HOA2D). Decoding into speaker signals is applied after a summation of the signals across all sources. In the decoder, the order gains can be configured to form a 'basic' decoder or a 'max $\mathbf{r}_E$' decoder [13]. A two-dimensional equal circular distribution of loudspeakers is assumed for this render format, because any other loudspeaker distribution would require a non-trivial decoding matrix. Although this render format applies principles of Ambisonics, it is a speaker-based render format, because encoding and decoding is combined.

All speaker based render formats use a max $\mathbf{r}_E$ first-order Ambisonics decoder for decoding of diffuse sound fields:

$$\mathbf{D} = \frac{1}{N} \begin{pmatrix} \sqrt{2} & g\tilde{s}_{1,x} & g\tilde{s}_{1,y} & g\tilde{s}_{1,z} \\ \vdots & \vdots & \vdots & \vdots \\ \sqrt{2} & g\tilde{s}_{n,x} & g\tilde{s}_{n,y} & g\tilde{s}_{n,z} \end{pmatrix}. \tag{20}$$

g is the decoder type dependent gain; for max $\mathbf{r}_E$ this is $g = 1/\sqrt{2}$ in the two-dimensional case and $g = 1/\sqrt{3}$ in the three-dimensional case [13].

2.3.3. Ambisonics-based render formats

First, second and third order render formats were implemented for three-dimensional rendering. They follow the Ambisonics channel number (ACN) sequence [57], using Furse-Malham normalization. The Ambisonics-based render formats encode plane waves, i.e., they do not account for near-field effects. For two-dimensional encoding, all output channels which are zero, $w_n \equiv 0$, are discarded.

2.3.4. Binaural rendering

Binaural signals and multi-channel signals for hearing aid microphone arrays $\hat{\mathbf{z}} = (\hat{z}_1, \ldots, \hat{z}_m)$ are generated by rendering to a virtual loudspeaker array, i.e., using a speaker-based render format, and applying a convolution of the loudspeaker signals z_n with the corresponding head-related impulse responses (HRIRs) $h_{n,m}$ for the respective loudspeaker directions. The HRIRs can be either recorded (e.g., [58, 59, 60]) or modeled [61]. This method is similar to the approach proposed by [46]; for real-time application of hearing aid algorithms, a Master Hearing Aid [44, 45] can be used. The reason for using virtual loudspeaker signals as an intermediate step in the binaural rendering is to allow for source movement as well as self-movement. Here, the render format can be seen as an interpolation method of the spatially sampled HRIR set. Self-movement can be simulated using a head tracker, and applying the movement to the receiver in the scene. This way, not only the listener's orientation, but also effects caused by translation, such as time-variant coloration or level changes of near sound sources, are accounted for.

3. Implementation

The implementation of TASCAR utilizes the Jack Audio Connection Kit [62], a tool for real-time audio routing between different pieces of software, and between software and audio hardware. The audio content is transferred between different components of the toolbox via JACK input and output ports. The JACK time-line is used as a base of all time-varying features, for data logging and as a link to the time-line of external tools. This is a major feature for simple setup of reproducible experiments, because audio content, simulated dynamic geometry information and the data logging of sensors, e.g., motion tracking, subject responses or bio-physical sensors always share the same time line. The same time line can also be accessed from external tools.

The audio signals are processed in blocks. The time-variant geometry and the dependent simulation coefficients, e.g., delay, air absorption filter coefficients or panning weights, are updated at the block boundaries. The simulation coefficients are linearly interpolated between the boundaries. This approximation by linear interpolation might be inaccurate if the simulation coefficients vary non-linearly within a block, e.g., panning weights during fast lateral movements.

Render formats and algorithmic trajectory generators are implemented as modules. Object properties, like geometry data, reflection properties and gains, and the time-line can be controlled interactively via a network interface, using OSC or LSL.

To achieve parallel processing in TASCAR, virtual acoustic environments can be split into multiple scenes. Independent scenes can be processed in parallel. Feedback signal paths, e.g., caused by external reverberation, are possible, but will lead to an additional block of delay. The delay and processing order of scenes is managed by the JACK audio back-end.

The sustainability of this software is granted by using publicly accessible source repositories. It is depending only on publicly available libraries. Furthermore, the software is continuously improved using professional software development methods, such as version management system, continuous integration tools, unit testing and system tests, as well as automated binary packaging and distribution methods. The proposed tool serves as a primary lab tool in several laboratories at the University of Oldenburg, and is used in publicly funded research projects, which guarantees an availability also in the future.

4. Example environment

A minimal example scene in TASCAR with one receiver, one sound source, and a "shoe-box" room is given in this section. A top-level TASCAR file contains a session, which can contain one or more scenes, and allows to load additional modules which are not directly related to the acoustic simulation, e.g., data logging, starting of additional tools, or loading of motion sensor interfaces. Each scene should contain at least one sound source and one receiver. Positions of objects are provided as a time series of Cartesian coordinates; here, the sound source with the name "source" is starting at $x = 1\,\mathrm{m}$, $y = -10\,\mathrm{m}$ at time zero, and is moving to $x = 1\,\mathrm{m}$ and $y = 10\,\mathrm{m}$ within 10 seconds. The audio content for the sound source is taken from a sound file, and is reproduced with a level so that the overall RMS level of the sound file would correspond to 70 dB SPL in an anechoic condition at a distance of 1 m. A "shoe-box" like room with the dimensions 5 m times 4 m times 3 m is used to simulate reflections. Trajectories and reflector meshes can be modeled in the 3D-authoring tool "blender" and exported into text files using helper functions of TASCAR.

```
<?xml version="1.0"?>
<session>
  <scene ismorder="2">
    <receiver name="out" type="omni"/>
    <source name="source">
      <position>0 1 -10  0
10 1 10 0</position>
      <sound>
        <sndfile name="myfile.wav" level="70"
         levelmode="rms"/>
```

```
      </sound>
    </source>
    <facegroup shoebox="5 4 3"
     reflectivity="1" damping="0.6"/>
  </scene>
</session>
```

Interaction with this scene is possible via MATLAB/GNU Octave with these TASCAR-provided scripts:

```
send_osc('localhost',9877,...
         '/scene/out/pos',1,0,0);
```

This will instantaneously shift the position of the receiver "out" by 1 m along the x-axis, in addition to the trajectory defined in the file. Continuous interaction, e.g., for head tracking, can be achieved by sending this kind of message upon each head tracking measurement.

Many more examples can be found in the software repository [33], including examples on the setup of diffuse reverberation with external tools.

5. Performance measurements

For a rough estimation of the factors that determine computational complexity in TASCAR, the CPU load was measured as a function of several relevant factors. The performance measurements were done with version 0.169 of TASCAR. All underlying render tools are part of the TASCAR repository [33].

5.1. Methods

CPU load C caused by audio signal processing was assessed using the 'clock()' system function, after processing 10 seconds of white noise in each virtual sound source. The number of primary sources K, number of output channels N, block size P, maximum length of delay lines l_d and the render format was varied (see Table I for an overview of the parameter space). For the 'HOA2D' render format, the maximum possible order $o = \frac{1}{2}(N-1)$ for the given number of output channels N was used. No image sources were processed, i.e., all simulated sources were omni-directional primary sources, and no reflectors were used during the performance measurements, because internally image sources are handled the same way as primary sources in terms of memory usage and computational complexity. In a typical scene, the number of image sources depends on the number and geometry of reflectors, the ISM order, and the number of primary sources – in a scene with a "shoe-box"-shaped room and a single primary source, the number of image sources increases from 6 for a first order ISM to 36, 162, 712, 3330 and 16088 for an ISM order of 2, 3, 4, 5 and 6, respectively. Each measurement of a combination of K, N, P, l_d and render format was repeated twice. The CPU load C is time per cycle τ_P in samples divided by length of cycle P in samples.

Number of sources and number of output channels are directly related to the numerical complexity in the receiver module. The block size controls the frequency of the geometry update. Memory usage is mainly affected by the maximum delay line length. One delay line is allocated in memory for each sound source. At 44.1 kHz sampling rate, the memory usage of the delay lines is 520 Bytes per meter and source. Different render formats may differ in their numerical complexity.

Table I. Parameter space of the performance measurements.

Factor	Values
number of sources K	1, 10, 100, 256
no. of output channels N	8, 48, 128
block size P	64, 256, 1024 samples
max. delay line length l_d	1 m, 10 km
render format	NSP, VBAP, HOA2D
CPU model	i5-2400@3.1GHz i5-6300HQ@2.3GHz i5-6500@3.2GHz i7-7567U@3.5GHz AMD FX-4300@3.8GHz AMD Ryzen 71700

5.2. Results

A one-way analysis of variances revealed that at all tested factors except for the delay line length and repetition showed a significant influence on the τ_P at a significance level of $p = 0.05$. Thus, in the following analysis the data was averaged across l_d and repetitions.

To provide an estimation of the contribution of different factors to the numerical complexity, a model function based on the implementation was fitted to the measured data:

$$\tau_P = \underbrace{a_0}_{\text{overhead}} + \underbrace{a_1 K}_{\text{geometry}} + \underbrace{a_2 K P}_{\text{source audio}} + \underbrace{a_3 N P}_{\text{postproc.}} + \underbrace{a_4 N K P}_{\text{panning}}. \quad (21)$$

In this model, a_0 represents the overhead by framework which is not related to the simulation properties. a_1 is an estimate of geometry processing time, which is performed for each source, but not depending on the number of audio samples per processing block P. The factor a_2 is related to source audio processing time per sample in the transmission model, and the processing time spent in the receiver, which does not depend on the number of loudspeakers. a_3 is an estimate of the post processing time per audio sample in the receiver, which does not depend on the number of sources. a_4 is time per audio sample for each loudspeaker and sound source, i.e., time spent in the panning function of the render format. The model parameters were found by minimizing the mean-square error between the measured and predicted CPU load C, and are shown in Table II. An example data set for one architecture and receiver type is shown in Figure 4.

It is often required to estimate the maximum number of sound sources K for a given CPU, render format and loud-

Table II. Results of the model fits of CPU load measurement.

CPU	format	a_0	a_1	a_2	a_3	a_4	$K_{\max,8}$	$K_{\max,48}$
i5-2400	NSP	0.045	0.017	0.001	0.00052	7.8e-05	541	182
@3.1GHz	VBAP	0.41	0.093	0.00036	0.00051	8.1e-05	812	201
	HOA2D	0.52	0.02	0.001	0.00088	4.1e-05	662	288
i5-6300HQ	NSP	0.028	0.0051	0.0011	0.00043	6.3e-05	548	210
@2.3GHz	VBAP	0.019	0.062	0.00059	0.00046	7e-05	742	220
	HOA2D	2.1e-06	0.016	0.0011	0.00069	3.7e-05	636	302
i5-6500	NSP	0.057	0.0034	0.001	0.00038	5.6e-05	615	238
@3.2GHz	VBAP	0.021	0.059	0.00053	0.00042	6.2e-05	825	246
	HOA2D	0.066	0.014	0.00098	0.00062	3.2e-05	714	341
i7-7567U	NSP	0.099	0.0046	0.00077	0.00036	4.6e-05	790	298
@3.5GHz	VBAP	0.036	0.071	0.00014	0.00033	5.2e-05	1443	329
	HOA2D	0.096	0.014	0.0008	0.00053	2.7e-05	868	410
AMD FX-4300	NSP	0.099	1.8e-09	0.00019	3.2e-05	0.00021	490	89
@3.8GHz	VBAP	1.4e-09	0.28	0.0012	3e-14	0.00017	316	93
	HOA2D	0.056	0.019	0.0016	0.0011	4.5e-05	441	221
AMD Ryzen 71700	NSP	1.1e-06	0.015	0.00087	0.00046	6e-05	661	234
@3.6GHz	VBAP	0.46	0.065	0.00027	0.00029	6.5e-05	1058	258
	HOA2D	0.064	0.016	0.00083	0.00061	3.6e-05	789	339

speaker setup (affecting N) and latency constraint (affecting P). Equation (21) can be transformed to

$$K_{\max} \leq \frac{C - a_0 P^{-1} - a_3 N}{a_1 P^{-1} + a_2 + a_4 N}. \tag{22}$$

As an example, $K_{\max}$ was calculated for all tested combinations of CPU model and receiver type, for $C = 90\%$ and $P = 1024$. These results are given in the last two columns of Table II, for $N = 8$ and $N = 48$.

The results show that on CPU models which are commonly used at the time of writing, several hundred sound sources can be simulated. From the tested render formats, 'HOA2D' was most efficient, especially for larger values of N. These results take only a single core into account. On multi-core computers, more complex environments can be simulated by splitting them into multiple scenes of lower complexity, and rendering them in parallel.

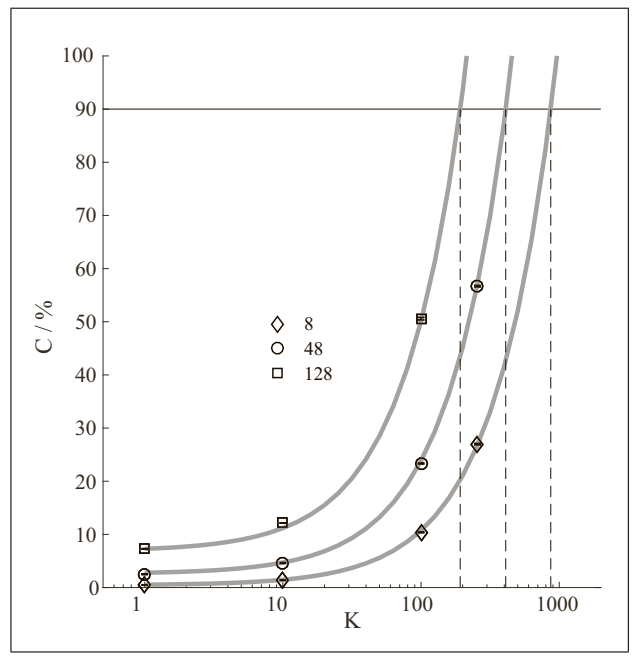

Figure 4. Example CPU load (i7-7567U@3.5GHz, HOA2D receiver, $P = 1024$): Measured data (symbols) with model fit (Equation (21), gray solid lines), for $N = 8$ loudspeakers (diamonds), $N = 48$ loudspeakers (circles) and $N = 128$ loudspeakers (squares). Vertical dashed lines indicate the maximum possible number of sources, Equation (22), for the given hardware.

6. Discussion

For hearing aid evaluation, providing a controllable complexity is an important factor [6]. The software tool TASCAR was developed for conducting hearing device evaluation in laboratories with a higher level of realism than typically applied. Therefore, features like moving sources and receivers, simulation early reflections in a geometric ISM, and interfaces for an efficient embedding into a laboratory environment were implemented. On the other hand, typical room acoustic modeling features, e.g., diffuse reverberation, which are available in other tools like RAVEN [28] or SOFE [31] are missing. Also in the acoustic model of TASCAR some approximations are applied. Most specifically reflection filters in the ISM are applied as a first-order lowpass filter only. In room-acoustic modeling this simplification can result in huge differences in simulation quality due to a huge number of bounces, e.g., resulting in an invalid reverberation time. However, in applications with low order ISMs in combination with external reverberation tools, the perceptual influence of the first-order ap-

proximation of reflection filters is low, as could be shown in [18]. Ideally, virtual acoustic environments for audiology would provide perfect authenticity, since it is yet unclear to what extent the signal processing of hearing devices and the perception of hearing impaired listeners is affected by the simplifications applied in rendering tools. With the proposed method we try to contribute to simulation tools for audiology by approaching it from the side of interactive dynamic rendering, with many interfaces to sensors and evaluation tools.

7. Summary and conclusions

In this technical paper, a toolbox for creation and rendering of dynamic virtual acoustic environments (TASCAR) was described, which allows direct user interaction. This tool was developed for application in hearing aid research and audiology. The three main modules of TASCAR – audio player, geometry processor and acoustic model – form the simulation framework. The audio player provides the tool with audio signals, the geometry processor keeps track of the distribution of the objects in the virtual space, and the acoustic model performs the room acoustic simulation and renders the scene into a chosen output format. The simulation uses a transmission model and a geometric ISM in the time domain, to allow for interactivity, and for a simple physical model of motion-related acoustic properties, such as Doppler shift and comb filtering effects. TASCAR allows selecting from a number of various rendering formats, customized to the needs of a range of applications, including higher order Ambisonics and binaural rendering formats. It offers a set of features, e.g., dynamic time-domain geometric ISM, diffuse sound field handling, directional sources, and interfaces for integration into laboratory environments, which is to current knowledge unique in this combination.

Performance measurements quantify the influence of factors related to simulation complexity. The results show that, despite some limitations in terms of complexity of the virtual acoustic environment, several hundred virtual sound sources can be interactively rendered, even over huge reproduction systems and on consumer-grade render hardware.

Acknowledgement

Work funded by DFG FOR1732 “Individualized Hearing Acoustics” and by the German Research Foundation DFG project number 352015383 – SFB 1330.

References

[1] V. Hamacher, J. Chalupper, J. Eggers, E. Fischer, U. Kornagel, H. Puder, U. Rass: Signal processing in high-end hearing aids: state of the art, challenges, and future trends. EURASIP Journal on Applied Signal Processing **2005** (2005) 2915–2929.

[2] T. Ricketts: Impact of noise source configuration on directional hearing aid benefit and performance. Ear and Hearing **21** (2000) 194–205.

[3] M. Cord, R. Surr, B. Walden, O. Dyrlund: Relationship between laboratory measures of directional advantage and everyday success with directional microphone hearing aids. Journal of the American Academy of Audiology **15** (2004) 353–364.

[4] R. A. Bentler: Effectiveness of directional microphones and noise reduction schemes in hearing aids: A systematic review of the evidence. Journal of the American Academy of Audiology **16** (2005) 473–484.

[5] V. Best, G. Keidser, J. M. Buchholz, K. Freeston: An examination of speech reception thresholds measured in a simulated reverberant cafeteria environment. International Journal of Audiology (2015) 1–9.

[6] G. Grimm, B. Kollmeier, V. Hohmann: Spatial acoustic scenarios in multichannel loudspeaker systems for hearing aid evaluation. Journal of the American Academy of Audiology **27** (2016) 557–566.

[7] B. Tessendorf, A. Bulling, D. Roggen, T. Stiefmeier, M. Feilner, P. Derleth, G. Tröster: Recognition of hearing needs from body and eye movements to improve hearing instruments. – In: Pervasive Computing. Springer, 2011, 314–331.

[8] B. Tessendorf, A. Kettner, D. Roggen, T. Stiefmeier, G. Tröster, P. Derleth, M. Feilner: Identification of relevant multimodal cues to enhance context-aware hearing instruments. Proceedings of the 6th International Conference on Body Area Networks, 2011, ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 15–18.

[9] G. Kidd Jr, S. Favrot, J. G. Desloge, T. M. Streeter, C. R. Mason: Design and preliminary testing of a visually guided hearing aid. The Journal of the Acoustical Society of America **133** (2013) EL202–EL207.

[10] M. De Vos, K. Gandras, S. Debener: Towards a truly mobile auditory brain–computer interface: exploring the p300 to take away. International journal of psychophysiology **91** (2014) 46–53.

[11] A. J. Berkhout, D. de Vries, P. Vogel: Acoustic control by wave field synthesis. The Journal of the Acoustical Society of America **93** (1993) 2764–2778.

[12] V. Pulkki: Virtual sound source positioning using vector base amplitude panning. J. Audio Eng. Soc **45** (1997) 456–466.

[13] J. Daniel: Représentation de champs acoustiques, application à la transmission et à la reproduction de scÃĺnes sonores complexes dans un contexte multimÃľdia. Dissertation. Université Pierre et Marie Curie (Paris VI), Paris, 2001.

[14] J. B. Allen, D. A. Berkley: Image method for efficiently simulating small-room acoustics. The Journal of the Acoustical Society of America **65** (1979) 943.

[15] T. Lentz, D. Schröder, M. Vorländer, I. Assenmacher: Virtual reality system with integrated sound field simulation and reproduction. EURASIP Journal on Applied Signal Processing **2007** (2007) 187–187.

[16] T. Wendt, S. Van De Par, S. D. Ewert: A computationally-efficient and perceptually-plausible algorithm for binaural room impulse response simulation. Journal of the Audio Engineering Society **62** (2014) 748–766.

[17] S. Bertet, J. Daniel, E. Parizet, O. Warusfel: Investigation on localisation accuracy for first and higher order ambisonics reproduced sound sources. Acta Acustica united with Acustica **99** (2013) 642–657.

[18] G. Grimm, J. Heeren, V. Hohmann: Comparison of distance perception in simulated and real rooms. Proceedings of the International Conference on Spatial Audio, Graz, 2015.

[19] G. Grimm, S. Ewert, V. Hohmann: Evaluation of spatial audio reproduction schemes for application in hearing aid research. Acta Acustica united with Acustica **101** (2015) 841–854.

[20] C. Oreinos, J. M. Buchholz: Objective analysis of ambisonics for hearing aid applications: Effect of listener's head, room reverberation, and directional microphones. The Journal of the Acoustical Society of America **137** (2015) 3447–3465.

[21] M. Lundbeck, G. Grimm, V. Hohmann, S. Laugesen, T. Neher: Sensitivity to angular and radial source movements as a function of acoustic complexity in normal and impaired hearing. Trends in Hearing **21** (2017).

[22] L. Savioja, J. Huopaniemi, T. Lokki, R. VÃď Ãď nÃď nen: Creating interactive virtual acoustic environments. J. Audio Eng. Soc **47** (1999) 675–705.

[23] Ease. http://ease.afmg.eu/.

[24] G. M. Naylor: OdeonâĂŤanother hybrid room acoustical model. Applied Acoustics **38** (1993) 131–143.

[25] A. Wabnitz, N. Epain, C. Jin, A. Van Schaik: Room acoustics simulation for multichannel microphone arrays. Proceedings of the International Symposium on Room Acoustics, 2010, Citeseer, 1–6.

[26] L. Picinali, A. Afonso, M. Denis, B. F. Katz: Exploration of architectural spaces by blind people using auditory virtual reality for the construction of spatial knowledge. International Journal of Human-Computer Studies **72** (2014) 393–407.

[27] J. Ahrens, S. Spors: An analytical approach to sound field reproduction using circular and spherical loudspeaker distributions. Acta Acustica united with Acustica **94** (2008) 988–999.

[28] D. Schröder, M. Vorländer: Raven: A real-time framework for the auralization of interactive virtual environments. Forum Acusticum, 2011, Aalborg Denmark, 1541–1546.

[29] M. Noisternig, B. F. Katz, S. Siltanen, L. Savioja: Framework for real-time auralization in architectural acoustics. Acta Acustica United with Acustica **94** (2008) 1000–1015.

[30] S. Favrot, J. M. Buchholz: LoRA: A Loudspeaker-Based room auralization system. Acta Acustica united with Acustica **96** (2010) 364–375.

[31] B. U. Seeber, S. Kerber, E. R. Hafter: A system to simulate and reproduce audio–visual environments for spatial hearing research. Hearing research **260** (2010) 1–10.

[32] G. Grimm, J. Luberadzka, T. Herzke, V. Hohmann: Toolbox for acoustic scene creation and rendering (tascar): Render methods and research applications. Proceedings of the Linux Audio Conference, Mainz, Germany, 2015, F. Neumann (ed.), Johannes-Gutenberg Universität Mainz.

[33] Tascar/gpl. https://github.com/HoerTech-gGmbH/tascar.

[34] Tascar/hörtech. http://www.tascar.org/.

[35] blender. http://www.blender.org/, 2013. Stichting Blender Foundation, Amsterdam, the Netherlands.

[36] M. Wright: Open sound control: an enabling technology for musical networking. Organised Sound **10** (2005) 193–200.

[37] C. Kothe: Lab streaming layer (lsl). https://github.com/sccn/labstreaminglayer. Accessed on May 25, 2018, 2018.

[38] C. Landone, M. Sandler: Issues in performance prediction of surround systems in sound reinforcement applications. Proceedings of the 2nd COST G-6 Workshop on Digital Audio Effects (DAFx99), Norwegian University of Science and Technology, Trondheim, Norway, December 1999.

[39] J. Daniel, R. Nicol, S. Moreau: Further investigations of high-order ambisonics and wavefield synthesis for holophonic sound imaging. Audio Engineering Society Convention 114, March 2003.

[40] K. Carlsson: Objective localisation measures in ambisonic surround-sound. Dissertation. Master Thesis in Music Technology, Supervisor: Dr. Damian Murphy. Department of Speech, Music and Hearing, Royal Institute of Technology, Stockholm. Work carried out at Dept. of Electronics University of York, 2004.

[41] V. Pulkki, T. Hirvonen: Localization of virtual sources in multichannel audio reproduction. Speech and Audio Processing, IEEE Transactions on **13** (2005) 105–119.

[42] E. Benjamin, A. Heller, R. Lee: Why ambisonics does work. Audio Engineering Society Convention 129, 11 2010.

[43] N. Epain, P. Guillon, A. Kan, R. Kosobrodov, D. Sun, C. Jin, A. van Schaik: Objective evaluation of a three-dimensional sound field reproduction system. Proceedings of the 20th International Congress on Acoustics (ICA 2010), Sydney, Australia, 2010, 23–27.

[44] G. Grimm, T. Herzke, D. Berg, V. Hohmann: The Master Hearing Aid – a PC-based platform for algorithm development and evaluation. Acta Acustica united with Acustica **92** (2006) 618–628.

[45] T. Herzke, H. Kayser, F. Loshaj, G. Grimm, V. Hohmann: Open signal processing software platform for hearing aid research (openMHA). Proceedings of the Linux Audio Conference, 2017, UniversitÃĭ Jean Monnet, Saint-ÃĽtienne, 35–42.

[46] F. Pausch, L. Aspöck, M. Vorländer, J. Fels: An extended binaural real-time auralization system with an interface to research hearing aids for experiments on subjects with hearing loss. Trends in hearing **22** (2018) 2331216518800871.

[47] M. M. E. Hendrikse, G. Llorach, G. Grimm, V. Hohmann: Virtual audiovisual everyday-life environments for hearing aid research. https://doi.org/10.5281/zenodo.1434116, Okt. 2018.

[48] U. P. Svensson, R. I. Fred, J. Vanderkooy: An analytic secondary source model of edge diffraction impulse responses. The Journal of the Acoustical Society of America **106** (1999) 2331–2344.

[49] A. Asheim, U. Peter Svensson: An integral equation formulation for the diffraction from convex plates and polyhedra. The Journal of the Acoustical Society of America **133** (2013) 3681–3691.

[50] J. Huopaniemi, L. Savioja, M. Karjalainen: Modeling of reflections and air absorption in acoustical spaces a digital filter design approach. Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA), 1997, IEEE.

[51] G. B. Airy: On the diffraction of an object-glass with circular aperture. Transactions of the Cambridge Philosophical Society **5** (1835) 283.

[52] A. J. Chadwick, S. Shelley: Openair lib impulse response database. http://www.openairlib.net/, 2015. Audio Lab, University of York.

[53] A. W. Bronkhorst, T. Houtgast: Auditory distance perception in rooms. Nature **397** (1999) 517–520.

[54] J. Merimaa, V. Pulkki: Spatial impulse response rendering. Proc. of the 7th Intl. Conf. on Digital Audio Effects (DAFXâĂŹ04), Naples, Italy, 2004.

[55] F. Zotter, M. Frank, M. Kronlachner, J.-W. Choi: Efficient phantom source widening and diffuseness in ambisonics. Proceedings of the EAA Joint Symposium on Auralization and Ambisonics, Berlin, 2014.

[56] J. Pätynen, S. Tervo, T. Lokki: Amplitude panning decreases spectral brightness with concert hall auralizations. Audio Engineering Society Conference: 55th International Conference: Spatial Audio, 2014, Audio Engineering Society.

[57] M. Chapman, W. Ritsch, T. Musil, J. Zmölnig, H. Pomberger, F. Zotter, A. Sontacchi: A standard for interchange of ambisonic signal sets. including a file standard with metadata. Proc. of the Ambisonics Symposium, Graz, Austria, 2009.

[58] H. Kayser, J. Anemüller, T. Rohdenburg, V. Hohmann, B. Kollmeier, et al.: Database of multichannel in-ear and behind-the-ear head-related and binaural room impulse responses. EURASIP Journal on Advances in Signal Processing (2009).

[59] J. Thiemann, A. Escher, S. van de Par: Multiple model high-spatial resolution hrtf measurements. Proceedings of the German annual conference on acoustics (DAGA), Nürnberg, 2015.

[60] F. Denk, S. M. Ernst, S. D. Ewert, B. Kollmeier: Adapting hearing devices to the individual ear acoustics: Database and target response correction functions for various device styles. Trends in hearing **22** (2018) 2331216518779313.

[61] R. O. Duda: Modeling head related transfer functions. Signals, Systems and Computers, 1993. Conference Record of The Twenty-Seventh Asilomar Conference on, 1993, IEEE, 996–1000.

[62] P. Davis, T. Hohn: Jack audio connection kit. Proceedings of the Linux Audio Developer Conference. ZKM Karlsruhe, 2003.