Browser Application for Virtual Audio Walkthrough

Thomas Deppisch
Student, Graz University of Technology
and University of Music and Performing Arts
Email: thomas.deppisch@student.tugraz.at

Alois Sontacchi
University of Music and Performing Arts
Institute of Electronic Music and Acoustics
Inffeldgasse 10, 8010 Graz, Austria
Email: sontacchi@iem.at

***Abstract*—We present an application allowing an interactive virtualization of auditory scenes. It enables the user to navigate through the virtual scene inside a web browser. Audio signals are spatialized for headphone playback using a binaural Ambisonics approach. A mixture of cues is used to activate and enhance distance perception. Customized scenes are created using a simple text file which contains meta data regarding properties of the virtual room and the audio objects. In order to scale the audio reproduction quality corresponding to available computational power, parameters like Ambisonics order and image source order are used to adjust the virtualization during runtime.**
The source code is provided online[1].

I. Introduction

Hitherto in conventional and classical audio recordings the acoustic perspective within the recording has been defined by the tonmeister. However, new developments [1] provide the possibility to follow new practices in media/audio immersion: Listeners can navigate throughout a production visiting any favored position of interest. The addressed invention [1] relates to an audio production, processing, and playback apparatus to convey a multichannel interactive audio experience, allowing the listener to traverse an entire sound scene. Hereinafter, we present a web based implementation of this approach.
Before going into implementation details, the following introduction states how direction and distance of acoustic sources are perceived and reproduced. Basic concepts of the Web Audio Application Programming Interface (API) for audio processing in a browser environment are shown as well.

A. Perception of direction

Cues for the perception of an acoustic source direction are classified into monaural and binaural cues [2]. Binaural cues utilize information from differences in both ear signals while monaural cues utilize equivalent parts of both ear signals to determine the direction of a sound source [2]. Binaural cues can be further divided into interaural level differences (ILDs) and interaural time differences (ITDs). ILDs arise due to head shadowing effects for signals with small wave lengths compared to the diameter of the head. Hence, lateral sources produce higher levels on the ipsilateral ear than on the contralateral ear [2]. The delayed arrival of a sound signal at the contralateral ear in comparison to the ipsilateral ear results in an interaural time difference. Such a delay is evaluated using the phase difference in both ear signals. For wave lengths smaller than the diameter of the head these phase differences do not contain useful information. Therefore, ITDs are predominantly used for localization of signals with low frequency content [2]. Still, evaluation of the signal envelope allows localization based on ITDs for higher frequency signal components [3].
Monaural cues are manifested in direction dependent spectral changes of the ear signals' frequency responses. These spectral changes emerge due to reflections on pinna and torso, resulting in constructive and destructive interferences. Spectral localization cues are predominantly important for localization of elevated sources in sagittal planes, to prevent confusions and ambiguities [2].

B. Head-related transfer function

Both, monaural and binaural cues are incorporated in the head-related transfer function (HRTF) and its time domain representative, the head-related impulse response (HRIR) [2]. The HRIR can be obtained by placing microphone probes inside the ear channels of a test person or dummy head and measuring the impulse response for a number of source directions [4]. The HRIR is generally direction-dependent and hence can be used to simulate direction of a source in binaural synthesis. For distances smaller than 1 m the HRTF also shows distance-dependent spectral variations. For non-static sources or when head movements are incorporated, interpolation of a finite number of measured HRTFs is essential [5]. The anthropometric differences between human individuals result in individual spectral differences in HRTFs which can lead to an impairment of the binaural experience when using non-individualized HRTFs.

C. Perception of distance

Distance perception for acoustic sources is generally less accurate than the perception of direction [6]. There are several acoustic cues which allow a distance estimation for sound sources but also non-acoustic cues that play a big role in overall distance perception. The most prominent acoustic distance cue is the inverse distance law for sound pressure which states a 6 dB reduction of sound pressure level when doubling the source distance in free field conditions [6].
Another acoustic distance cue is the direct-to-reverberant energy ratio in reflective environments. Here, close sources provide a greater amount of direct energy in comparison to reverberant energy [6]. For sources further away than 15 m, air

[1] https://git.iem.at/thomasdeppisch/walkthrough

absorption results in high frequency attenuation and therefore in spectral distance cues [2]. Furthermore, for sources closer than 1 m an increase in low frequency ILDs has a strong impact on distance perception for close sources [7].

D. The Web Audio API

The Web Audio API[2] (WAA) allows modular audio processing in a web browser environment. Audio signals thereby are sent through an audio routing graph consisting of audio nodes which can be connected arbitrarily. A source node such as the *MediaElementAudioSourceNode* allows the integration of audio files into the routing graph. Several predefined audio nodes such as *BiquadFilterNode*, *DelayNode*, *GainNode* and *ConvolverNode* provide the possibility of realtime audio processing. The *AudioDestinationNode* connects the audio routing graph to the audio hardware. The WAA also allows basic spatialization by providing a *SpatialListenerNode* and a *SpatialPannerNode*. Customization of settings like HRTF set, distance function and directivity function are currently not possible [8].

II. Related work

So far, traversing a sound scene in reproduction could be realized by audio spatialization based on isolated recordings combined with additional spatial recordings or rendering of reverberation (object-based). Although the listener is meant to be located at a central position, by changing the arrangement of the virtual sources the playback perspective at the reproduction side can be adapted. There are several products allowing the use of this approach, e.g. Fraunhofer Spatial Sound Wave[3], or the Ambix Plugin Suite [9].

Moreover, Pihlajamäki and Pulkki [10] presented a different approach based on the DirAC [11] method. There the sound field is decomposed into a non-diffuse and diffuse part. Then the non-diffuse part gets resynthesized by assigning a direction to each frequency band. Transformations of the direction vectors, gain control and diffuseness control are used to simulate translations of the listener.

A method for sound field navigation using Ambisonics was presented by Allen and Kleijn [12]. After the directional decomposition of a signal, an adjustment for the translated origin is performed by filtering. Re-encoding is done in respect to the new angles based on the translation vector.

BogJS is a JavaScript framework for object-based audio rendering in browsers[4]. A demo[5] shows the use case of auditory scene virtualization in a web browser. As the spatialization is done solely with Web Audio API functionalities, the possibilities of personalization (e.g. change of the HRTF set) and flexible adjustments (e.g. of the distance gain function) are restricted.

[2]https://www.w3.org/TR/webaudio/

[3]https://www.idmt.fraunhofer.de/en/institute/projects_products/q_t/spatialsound_wave.html

[4]https://github.com/IRT-Open-Source/bogJS

[5]https://lab.irt.de/demos/object-based-audio/interactive/

III. Recording an auditory scene for virtualization

For recording of auditory scenes with the goal of later virtualization two approaches are feasible: Virtualization of sound objects recorded through spot microphones or virtualization of the scene recorded by multichannel microphone arrays (cf. figure 1). In the first case every microphone signal represents an acoustic object in the virtual space, e.g. a musical instrument. In the second case the signals of one microphone array represent a part of the sound field spatially sampled at one point in the room. Hence, the overall sound intensity of the multichannel microphone arrays needs to be normalized, so a higher density of microphone arrays in one part of the room does not result in a higher intensity. A hybrid approach combining spot microphones and multichannel microphone arrays is also feasible. During playback every microphone capsule is interpreted as a virtual speaker object which then gets placed in the room according to its original position.

Fig. 1. Recording an auditory scene using multichannel microphone arrays.

IV. Binaural synthesis using a virtual Ambisonics approach

A. Ambisonics

By solving the Helmholtz equation a spherical harmonics transform is obtained. Applying the spherical harmonics transformation to a point source leads to Ambisonics encoding (cf. eq. (1)) and decoding equations (cf. eq. (2)) [13].

$$\vec{\chi}_N(t) = \vec{y}_N(\vec{\theta}_0)s(t) \tag{1}$$

$$\vec{s}_{ls}(t) = \mathbf{D}\ \mathrm{diag}\{\vec{a}_N\}\ \vec{\chi}_N(t) \tag{2}$$

Multiplication of the signal $s(t)$ with the spherical harmonics evaluated at the desired source position $\vec{\theta}_0$ contained in $\vec{y}_N$, yields the Ambisonics encoded signals $\vec{\chi}_N(t)$ (eq. (1)). The order at which the evaluation of spherical harmonics is truncated is called Ambisonics order N. The encoded signals are decoded to speaker signals $\vec{s}_{ls}(t)$ by multiplication with a suitable decoder matrix $\mathbf{D}$ (eq. (2)). The decoder matrix can be obtained in several ways such as mode-matching, sampling or AllRAD [14]. The vector $\vec{a}_N$ can contain psychoacoustically motivated optimization factors, e.g. for *max* $\vec{r}_E$ optimization. *Max* $\vec{r}_E$ optimization reduces sidelobes and therefore leads to

a more distinct source localization (cf. [14], [15]).
Apart from full periphonic (3D) Ambisonics, circular harmonics can be employed to obtain planar (2D) Ambisonics. Further, mixed-order schemes are used to encode horizontal source information in higher order than vertical information [16]. Rotation of a sound field is done efficiently in the Ambisonics domain by matrix multiplication as described in [17].

B. Virtual Ambisonics approach

For binaural synthesis a virtual Ambisonics approach is used [18]. A regular distribution of virtual speakers is placed around the virtual listener. Decoding of the encoded Ambisonics signals $\vec{\chi}_N(t)$ at the virtual speaker positions $\vec{\theta}_q$ is achieved by multiplication with the decoding matrix $\mathbf{D}_{vls}$. The binaural signals for the left and right ear (eq. (3), (4)) are obtained by convolving the resulting virtual loudspeaker signals with their corresponding HRIRs and summing them up for each ear.

$$s_l(t) = \sum_{q=1}^{m} HRIR_{l,q}(\vec{\theta}_q) * (\vec{e}_q^T \ \mathbf{D}_{vls} \ \vec{\chi}_N(t)) \tag{3}$$

$$s_r(t) = \sum_{q=1}^{m} HRIR_{r,q}(\vec{\theta}_q) * (\vec{e}_q^T \ \mathbf{D}_{vls} \ \vec{\chi}_N(t)) \tag{4}$$

This approach, in contrast to HRTF interpolation methods [5], allows a rotation of the encoded sound field in the Ambisonics domain [17] instead of interpolation of HRTFs for every sound object. Therefore, the number of needed HRTFs is only depending on the number of virtual speakers and not on the number of virtual sound objects. This can reduce the amount of convolutions needed and hence reduce the computational effort.

V. Implementation

A. General functionality

The application uses an interaction of JavaScript code and Web Audio API (WAA) audio nodes based on C++ implementations. Background signal processing such as convolutions, filtering and gain adjustments are accomplished by WAA audio nodes. The calculations to retrieve the values for spatialization are done in JavaScript code. For Ambisonics processing, classes of the open source JavaScript library JSAmbisonics[6] [19] were adapted to provide periphonic as well as planar Ambisonics processing. As JSAmbisonics is built on top of the WAA as well, a seamless integration is possible.
In the following the construction of auditory scenes allowing a virtual walkthrough is explained step by step.

B. Scene File

To construct the virtual scene, meta data needs to be provided in a simple text file, the scene file (cf. figure 2). A valid scene file needs to follow the JSON[7] (JavaScript Object Notation) standard.

[6]https://github.com/polarch/JSAmbisonics
[7]http://json.org/

```
[{
        "type": "room",
        "width": 4.5,
        "length": 5.5,
        "height": 4,
        "listenerStart": {"x":2,"y":1}
},{
        "type": "mono",
        "name": "Noise",
        "position": {"x":1,"y":1,"z":1},
        "gain": 0.8,
        "NFC" : 1,
        "orientation": {"azim":90,"elev":-45},
        "distGain": {"a": 1.4,"g0": 1},
        "file": "sounds/noise.wav"
},{
        "type": "fourChannelArray",
        "name": "Oktava",
        "center": {"x":4,"y":4},
        "centerDistance": 0.5,
        "directivity": 0.5,
        "file": "sounds/oktava1.ogg",
        "channelMapping": {"speaker1":1,"speaker2":2,"
             speaker3":3,"speaker4":4}
}]
```

Fig. 2. Example for a valid scene file containing scene meta data.

In the first section the scene file provides information of the room as well as coordinates for the starting point of the virtual listener. Below the room data an arbitrary number of audio objects can be defined. Objects of type *mono* are based on a mono audio track, e.g. a spot microphone recording. Objects of type *fourChannelArray* represent a spatially sampled part of the sound field recorded by a microphone array consisting of four capsules. Each defined audio object has parameters like position, gain, orientation, distance gain function, directivity and reference to a sound file. Optionally, near field compensation filters (NFC) which approximate the filters given in [20] can be activated for mono objects. Objects of type *fourChannelArray* are defined by a center position and a center distance for each of the four corresponding virtual speakers. The sound file of a *fourChannelArray* object contains four separate mono channels. To map these four mono channels to the corresponding virtual speaker object, a channel mapping parameter is provided.
By using the directivity parameter a virtual speaker radiation directivity can be controlled. The directivity gain follows equation (5) and hence enables interpolation between omnidirectional ($\gamma = 1$), cardioid ($\gamma = 0.5$) and figure of eight ($\gamma = 0$). These directivity patterns are also valid for the three dimensional space as the angle φ is calculated as the angle between the vector pointing from the virtual speaker to the listener and the vector pointing in the same direction as the speaker.

$$g_{dir} = \gamma + (1 - \gamma) cos(\varphi) \tag{5}$$

The distance gain function follows equation (6) and can be adjusted by using the parameters α and g_0. The resulting distance gain equals 1 for a distance $r = 1$, linearily interpolates to g_0 for distances $r < 1$ and decreases by $1/r^\alpha$ for distances $r > 1$.

$$g_{dist} = \begin{cases} g_0 + (1 - g_0)r & \text{, if } r \leq 1 \\ \frac{1}{r^\alpha} & \text{, if } r > 1 \end{cases} \tag{6}$$

Figure 3 shows the default distance gain functions for objects of type *mono* and type *fourChannelArray*. For far distances the default distance gain decreases by $1/r^{1.4}$. This is an overproportional decrease compared to the inverse distance law and responds to the fact that physical distance is generally rather underestimated by humans when sound intensity is the primary cue [6]. For close distances the default gain depends on the object type: For *fourChannelArray* objects which do not represent an actual audio source but a part of the sound field, the distance gain decreases for close distances, so a single virtual speaker does not get too prominent. For *mono* objects the gain remains constant at $g_{dist} = 1$.

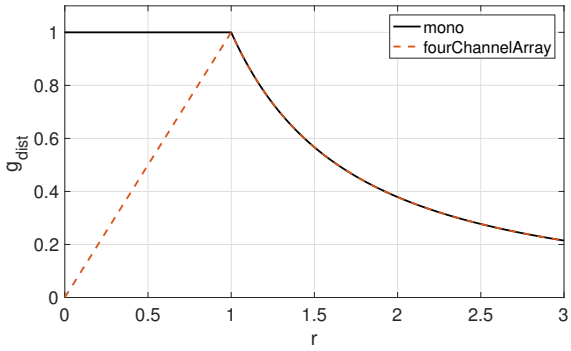

Fig. 3. Default distance gain as a function of distance r for objects of type *mono* (solid black) and type *fourChannelArray* (dashed red).

C. Virtualizing the scene

In the next step, when activated, mirror image sources for each audio object are built. These additional copies of sources follow the concept of simulating room reflections by mirroring sources along the room boundaries as explained in [21]. Image sources of first and second order are provided in the application and can be activated during runtime. Activation of image sources in big auditory scenes can lead to performance impairments due to the fact that the number of sources and hence all calculations for spatialization are multiplied.

From this point onwards the signal processing steps are displayed in a block diagram (cf. figure 4). Relating to the position of the virtual listener, angle and distance to each virtual speaker are calculated dynamically. From this data directivity gain and distance gain as described in eqations (5), (6), as well as a dynamic delay line (equation (7)) are adjusted.

$$\Delta t = \frac{r}{343\frac{m}{s}} \tag{7}$$

As the delay is adjusted dynamically to fit the distance r between listener and speaker, it is able to reproduce the Doppler shift.

Image sources are then lowpass filtered simulating a high frequency loss caused by absorption during reflections on room boundaries. In the last step before Ambisonics encoding, loudspeaker objects corresponding to *fourChannelArray*

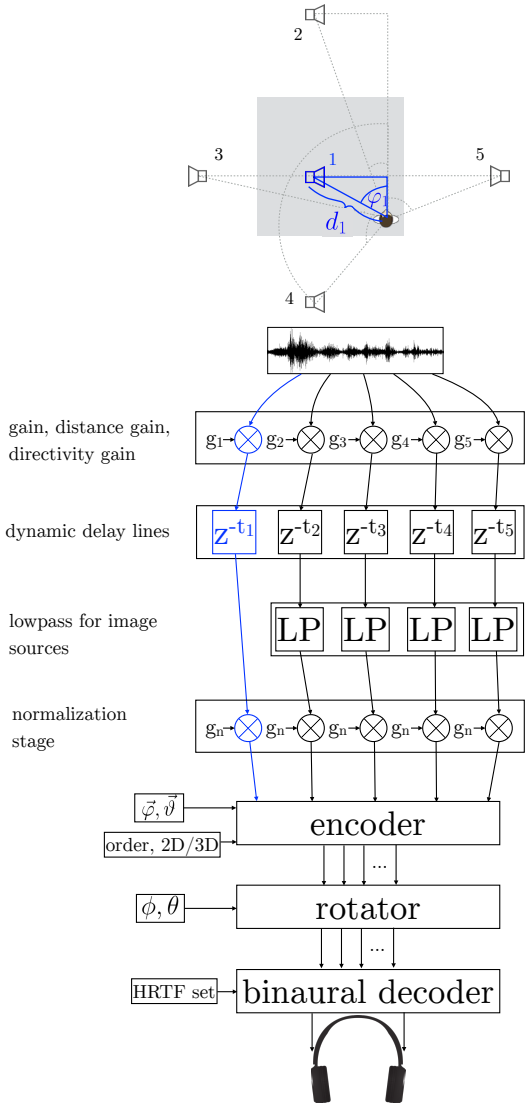

Fig. 4. Signal processing block diagram.

sources are intensity normalized as described in section III.

Before binaural headphone signals are obtained, Ambisonics encoding, rotation and decoding takes place. The encoder evaluates spherical or circular harmonics at the speaker directions relative to the virtual listener. *Mono* as well as *fourChannelArray* sources are encoded in an adjustable Ambisonics order N. If *fourChannelArray* sources (first order Ambisonics microhones) are encoded in a higher order than first order, the sound field does not get reproduced accurately. Yet, due to the superposition of several sound field sample points, audio information from the four room directions get reproduced more sharply when higher order encoding is enforced. For close distances all Ambisonics channels but the W-Channel (contains omnidirectional information) get interpolated to zero to avoid discontinuities when passing through a virtual speaker object.

The Ambisonics rotator is able to rotate the whole sound field in the Ambisonics domain. It enables head rotations of the virtual listener.
At the decoding stage Ambisonics signals get decoded to a regular distribution of virtual speakers. The number of virtual speakers depends on the Ambisonics order N: For periphonic Ambisonics a t-design [22] of degree $t = 2N+1$ and for planar Ambisonics a circular distribution of $2N + 2$ speakers are used. For HRTF individualization arbitrary SOFA[8] (Spatially Oriented Format for Acoustics) HRTFs are supported.

D. User interface

Figure 5 shows the user interface of the application. Before scene playback can be started, an auditory scene and an HRTF set need to be chosen by using the blue dropdown menus. Optionally, the Ambisonics type, Ambisonics order N and image source order can be adjusted to fit the scene-specific needs and computational possibilities. The Ambisonics type can be switched between 2D, 3D and 2D in combination with first order 3D components. The restriction of a maximum of 32 channels per audio node by the WAA and a highest supported *t-design* of degree $t = 21$ by JSAmbisonics yields maximum Ambisonics orders of $N = 4$ for 3D, $N = 15$ for 2D and $N = 10$ for 2D with first order 3D components. The navigation of the listener (depicted by a head, cf. figure 5) is accomplished by mouse dragging or using the up and down arrow keys. The left and right arrow keys as well as the azimuth slider are used to turn the head of the listener in the horizontal plane. The elevation slider is used to perform up and down head movements which are not graphically depicted as the scene is represented from a 2D perspective. Alternatively, head movements can be controlled via a low-cost open-source MIDI headtracker[9] [23] which is integrated using the Web MIDI API[10] and WebMidi.js[11]. The usage of a headtracker is currently only possible either using Chrome or Opera browsers, supporting the Web MIDI API. A volume slider and a start/pause toggle allow controlling the playback. A grey canvas below the settings section represents the room. Inside the canvas the listener and the virtual speaker objects are depicted. A virtual speaker object of type *mono* is depicted by a single speaker symbol. Virtual speaker objects of type *fourChannelArray* are represented by four speaker symbols arranged in a circle.

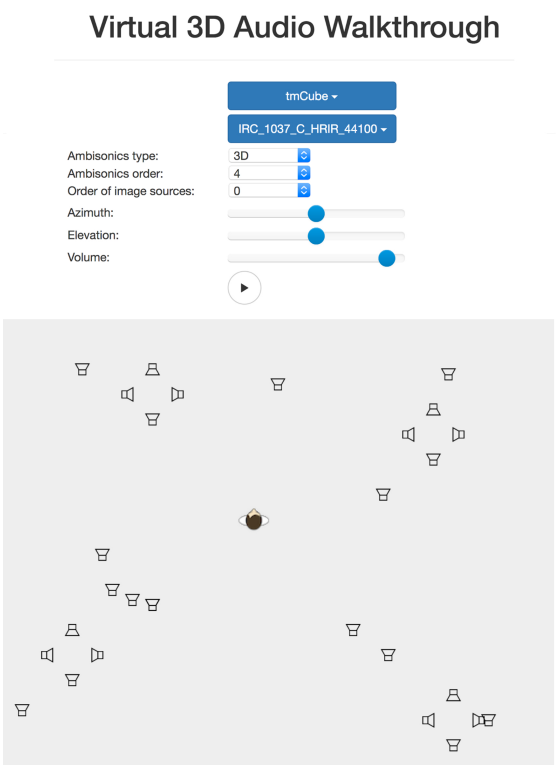

Fig. 5. User interface of the application.

VI. Conclusion and Outlook

After informal listening tests the presented application creates a promising impression of a virtual concert scene. Localization of single sound sources works well, especially if *mono* sources (corresponding to spot microphones at recording stage) are used. The use of *fourChannelArray* sources (corresponding to microphone arrays sampling a part of the sound field) enhances the immersion. Therefore, a combination of *fourChannelArray* and *mono* sources leads to the best results. Crosstalk between spot microphones should be avoided as much as possible as it may split the perceived direction of a sound source. The perceived immersion due to a valid room impression can be further improved by using image sources. Unfortunately, for big auditory scenes it is often not possible to activate image sources as the number of simultaneously processed audio channels rises by a multiple for every image source order. For big auditory scenes containing a high number of audio channels the computational power of an average personal computer may then be insufficient. The efficiency of the program might be improved by using an underlying C++ implementation integrated through a JavaScript wrapper like in WAA audio nodes. New drafts of the WAA also contain *AudioWorkerNode* classes which might be able to enhance the performance. Further challenges occur when embedding the application into a website: The limited download speed might prohibit the playback of big auditory scenes due to the big amount of audio data which needs to be downloaded.

[8]https://www.sofaconventions.org/
[9]https://git.iem.at/DIY/MrHeadTracker
[10]https://www.w3.org/TR/webmidi/
[11]https://github.com/cotejp/webmidi

References

[1] "Method and apparatus acoustic scene playback," European patent application: PCT/EP2016/075595, applicant institution: HUAWEI Technologies CO. LTD. (China), inventors: Schörkhuber Christian, Zotter Franz, Frank Matthias, Höldrich Robert, and Grosche Peter.
[2] J. Blauert, *Spatial Hearing: The psychophysics of human sound localization*. MIT Press, 1997.
[3] E. Macpherson and J. Middlebrooks, "Listener weighting of cues for lateral angle: The duplex theory of sound localization revisited," *Journal of the Acoustical Society of America*, vol. 111, no. 5, 2002.

[4] H. Moller, M. F. Sorensen, D. Hammershi, and C. B. Jensen, "Head-related transfer functions of human subjects," *J. Audio Eng. Soc*, vol. 43, no. 5, pp. 300–321, 1995.

[5] K. Hartung, J. Braasch, and S. J. Sterbing, "Comparison of different methods for the interpolation of head-related transfer functions," in *AES 16th International Conference on Spatial Sound Reproduction*, 1999.

[6] P. Zahorik, D. Brungart, and A. Bronkhorst, "Auditory distance perception in humans: A summary of past and present research," *Acta Acustica united with Acustica*, vol. 91, pp. 409–420, 05 2005.

[7] D. S. Brungart, N. I. Durlach, and W. M. Rabinowitz, "Auditory localization of nearby sources. II. Localization of a broadband source," *The Journal of the Acoustical Society of America*, vol. 106, no. 4, pp. 1956–1968, 1999.

[8] T. Carpentier, "Binaural synthesis with the Web Audio API," in *1st Web Audio Conference (WAC), Paris, France*, 2015.

[9] M. Kronlachner, "Ambisonics plug-in suite for production and performance usage," in *Linux Audio Conference*, 2013.

[10] T. Pihlajamäki and V. Pulkki, "Synthesis of complex sound scenes with transformation of recorded spatial sound in virtual reality," *J. Audio Eng. Soc*, vol. 63, no. 7/8, pp. 542–551, 2015.

[11] V. Pulkki, "Spatial sound reproduction with directional audio coding," *J. Audio Eng. Soc*, vol. 55, no. 6, pp. 503–516, 2007.

[12] A. Allen and W. B. Kleijn, "Ambisonic soundfield navigation using directional decomposition and path distance estimation," in *4th International Conference on Spatial Audio, Graz, Austria*, 2017.

[13] F. Zotter, "Analysis and synthesis of sound-radiation with spherical arrays," Dissertation, University of Music and Performing Arts, Graz, A, 2009.

[14] F. Zotter and M. Frank, "All-round ambisonic panning and decoding," *J. Audio Eng. Soc., Vol. 60, No. 10*, 2012.

[15] M. Gerzon, "General metatheory of auditory localisation," in *Preprint 3306, 92nd Conv. Audio Eng. Soc.*, 1992.

[16] C. Travis, "A new mixed-order scheme for ambisonic signals," in *Ambisonics Symposium*, Graz, A, 2009.

[17] J. Ivanic and K. Ruedenberg, "Rotation Matrices for Real Spherical Harmonics. Direct Determination by Recursion," *The Journal of Physical Chemistry*, vol. 100, no. 15, pp. 6342–6347, 1996.

[18] M. Noisternig, T. Musil, A. Sontacchi, and R. Höldrich, "3d binaural sound reproduction using a virtual ambisonic approach," in *IEEE International Symposium on Virtual Environments, Human-Computer Interfaces and Measurement Systems*, 2003.

[19] A. Politis and D. Poirier-Quinot, "JSAmbisonics: A Web Audio library for interactive spatial sound processing on the web," *Interactive Audio Systems Symposium, York, UK*, 2016.

[20] J. Daniel and S. Moreau, "Further study of sound field coding with higher order ambisonics," in *116th Conv. Audio Eng. Soc.*, 2004.

[21] J. Allen and D. Berkley, "Image method for efficiently simulating small-room acoustics," *Journal of the Acoustical Society of America*, vol. 65, no. 4, pp. 943–950, 1979.

[22] F. Zotter, M. Frank, and A. Sontacchi, "The virtual t-design ambisonics-rig using vbap," in *1st EAA Euroregio Ljubljana*, 2010.

[23] M. Romanov, P. Berghold, D. Rudrich, M. Zaunschirm, M. Frank, and F. Zotter, "Implementation and evaluation of a low-cost head-tracker for binaural synthesis," in *Paper 9689, 142nd Conv. Audio Eng. Soc.*, 2017.