Personalization support for binaural headphone reproduction in web browsers

Michele Geronazzo
Dept. of Information Engineering
University of Padova
Padova, Italy
+39 049 827 6976
geronazzo@dei.unipd.it

Jari Kleimola
Dept. of Media Technology
Aalto University
Espoo, Finland
jari.kleimola@aalto.fi

Piotr Majdak
Acoustics Research Institute
Austrian Academy of Sciences
Vienna, Austria
+43 1 51581 2511
piotr@majdak.com

ABSTRACT

This study considers the issue of providing an individual listening experience for binaural sound reproduction in web browsers via headphones. The proposed solution aims at building a web framework with Web Audio API, giving support to the download of head-related transfer functions (HRTFs) associated with listener's personal profile from a server and the synchronization between the listener's devices. With each playback device and listener, the individual headphone equalization filters will be computed from headphone transfer functions (HpTFs) stored on the server. At server side, we propose to store the HRTFs and HpTFs in spatially oriented format for acoustics (SOFA). At client-side, we propose to convert the data to a new structure (WAV) ensuring a compatible solution with existing Web Audio API implementations. A binaural rendering implementation in JavaScript acting as a proof-of-concept reveals critical issues related to the native implementation in web browsers.

Categories and Subject Descriptors

H.3.5 [**Information Storage And Retrieval**]: Online Information Services - Web-based services.
H.5.1 [**Information Interfaces and Presentation (e.g. HCI)**]: Multimedia Information Systems - *Artificial, augmented, and virtual realities.*
H.5.5 [Information Interfaces and Presentation (e.g. HCI)]: Sound and Music Computing - *Signal analysis, synthesis, and processing.*

General Terms

Algorithms, Performance, Human Factors, Standardization.

Keywords

spatial audio, head-related transfer function, individualization, web application.

1. INTRODUCTION

Given an arbitrary sound scene and room by placing an array of microphones in a specific location, it is ideally possible to reproduce the recorded sound field on a fixed array configuration of loudspeakers and virtually render virtual sound sources on top of it. This acoustic installation can be used by anyone, however the listener's movements are restricted to a small sweet-spot and a specific location where the installation is placed. The future of 3D-media environments and broadcasting is constantly moving towards high spatial fidelity in sound field reproduction systems that are able to overcome such limitations. One of this research directions consists of the personal fruition of spatialized audio contents, leading to the next generation of portable audio devices. Audio technologies should be fully integrated with mobile devices and browsers, being able to provide an authentic individual listening experience.

Listener's perception of auditory events spans three main groups of perceptual attributes, or *elemental senses* [1]: temporal attributes (rhythm, durability, reverberance, etc.), quality attributes (loudness, pitch, timbre, etc.), and spatial attributes (direction, distance, spatial impression, etc.). This paper focuses on the latter group of elements, relying on the auditory cues provided by the human body, i.e. binaural cues, such as *interaural level difference* (ILD) and *interaural time differences* (ITD), and monaural cues, such as the spectral coloration caused by external ear filtering, which are summarized into the so-called *Head-Related Transfer Functions* (HRTFs). HRTFs naturally filter sounds in free field and sounds presented via loudspeakers, e.g. a stereo or a 5.1 loudspeaker system. When HRTFs are known, they can be used to simulate that filtering for the presentation of virtual sound sources via headphones. Such a technique is usually called binaural synthesis, and the whole apparatus can be called *virtual auditory display* (VAD) [2]. In a virtual auditory display, for each virtual sound source, the signal is filtered by the pair of HRTFs corresponding to the spatial direction of the source [3] and a pair of *Headphone Transfer Functions* (HpTFs) counting for the acoustic properties of the playback device worn by the listener. Listener-specific HRTFs and HpTFs are required in order to provide sound localization performance equivalent to that for free field [4].

Providing to the listener his/her own measured individual HRTF/HpTF data is currently not manageable due to the need of specific hardware and anechoic spaces. Nowadays, most of VADs employ non-individual (or generic) HRTFs, usually measured on anthropomorphic mannequins, resulting in frequent localization errors and uncomfortable listening experiences.

The lack of individualization is one of the main issues that hinders the diffusion of binaural audio technologies even if spatial audio rendering with generic HRTFs is already supported in many audio

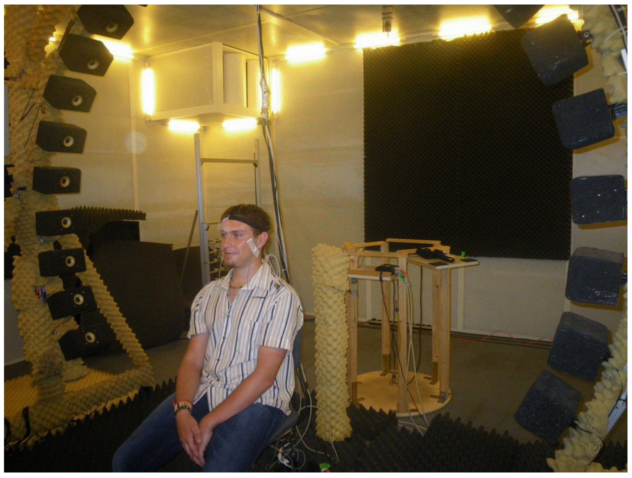

Figure 1: HRTF measurement setup at the Acoustics Research Institute, Austrian Academy of Sciences, Vienna.

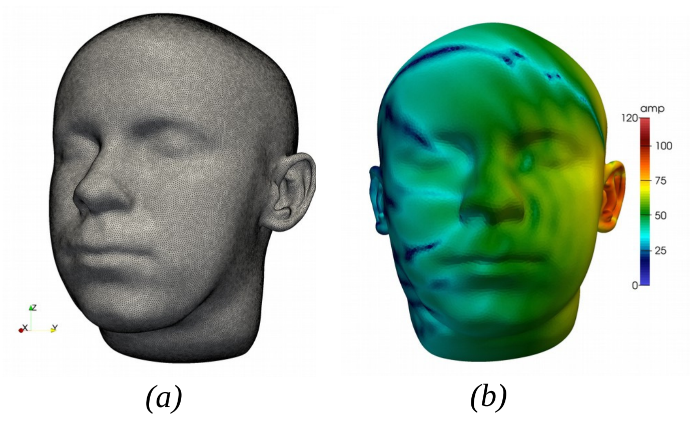

Figure 2: (a) Mesh of a human head and (b) acoustic pressure from a BEM calculation used in HRTF simulations.

software, e.g. OpenAL[1], Cycling '74 Max/MSP[2] and Pure Data[3], and recently incorporated in browsers via Web Audio API inside WebKit[4] and Firefox[5]. In light of such existing technologies, the main motivation behind this work is to promote personalization support in browsers and technologies for HRTF/HpTF personalization and customization which are able to fit listener needs world wide.

The *PannerNode* object in Web Audio API is able to perform HRTF rendering with a predetermined HRTF set, leading a generalization process of such object instead of developing a new custom object. The proposed solution is a first step in the standardization process for the research community. A client-side implementation in JavaScript would suggest guidelines and critical issues related to native implementation in browsers.

The main contributions of this paper are threefold. First, the proposed solution promotes the spatially oriented format for acoustics (SOFA) [5] in order to support personalized HRTFs/HpTFs in browsers. Second, a web browser integration and standardization should take into account the widespread WAV audio format for user profiling which is already supported and compatible. Finally, many technologies are already available to support the personalization process and they are discussed here.

2. PERSONALIZED 3D AUDIO

2.1 Spatial audio over headphones

Headphones allow to listen to binaural signals. A binaural signal is created by filtering a desired monophonic and anechoic sound signal with a pair of HRTFs. By using listener-specific HRTFs for the filtering, one can reach almost the same localization accuracy as in free-field listening conditions. When head motion, artificial reverberation and *room impulse responses* (RIRs) are involved even non-individualized HRTFs can provide a good sound localization performance [6].

[1] http://www.openal.org
[2] http://cycling74.com/products/max/
[3] http://puredata.info/
[4] http://www.webkit.org
[5] https://wiki.mozilla.org/WebAudio_API_Rollout_Status

When the binaural signals are adequately compensated for headphone-induced spectral coloration, the listeners are not able to distinguish between a real and a virtual sound source [7]. The transfer function between headphone and eardrum, described by the HpTFs, heavily varies from person to person and with small displacements of the headphone itself. The headphone-induced spectral coloration exhibits a limited inter-subject variability up to ≈ 4 kHz, because headphones act as an acoustic cavity only introducing a constant level variation. In the higher frequencies, headphone position with respect to listener’s ears introduces several deeps and valleys in the spectrum [8].
An inaccurate and non-individual compensation likely leads to spectral coloration that might affect the spatial audio fidelity. Although various techniques have been proposed in order to tackle such a delicate issue, modeling the correct equalization filter is still a hot open research theme.

2.2 Modeling HRTFs

The acoustic measurements of HRTFs are done by inserting a microphone into the ear canal of a listener and performing electro-acoustic system identification. Figure 1 depicts an HRTF measurement setup, where hundreds of directions are usually considered in time-consuming procedure lasting for tens of minutes. Thus, several attempts have been proposed to speed-up the measurements for human listeners [9].
HRTFs can also be numerically simulated. The simulations can be based on empirical approaches where parameters of a structural model are determined [10], [11]. The simulations can also be based on numerical calculations of the sound pressure based on the numeric representation of listener's geometry [12]. For the latter approach, boundary-element methods (BEM) [13] and finite-difference time-domain methods [14] have been successfully applied. In both approaches, an accurate representation of the listener's geometry is required (Fig. 2). Thus, further work on enabling HRTF simulations available for the masses include the improvements on geometry capturing [15], [16] and speeding up the calculations [17].
A novel framework for synthetic HRTF design and customization combines the structural modeling paradigm with selection techniques of generic HRTFs. The Mixed Structural Modeling (MSM) [18] approach treats HRTFs as a combination of structural components, which can be either synthetic or measured.

Each component is customized on individual anthropometric data, which are employ to generate model parameters or to select an available measured/simulated HRTF in a database [19].

3. SYSTEM ARCHITECTURE

3.1 Spatialization with Web Audio API

Web Audio API renders 3D spatial audio based on sound sources and their relation to a singleton *AudioListener* attribute of *AudioContext*. The sound sources are represented as *PannerNode* instances, and parametrized using 3D position, orientation, and velocity vectors, and attributes defining the directionality and the distance model of the sound source. The *AudioListener* interface holds listener's position and orientation in 3D audio scene, and encodes general properties of the scene itself. The implementation details of the algorithm are described in Section 3.3.

In WebKit, the measured impulse responses stem from a hybrid set extracted from IRCAM's LISTEN database[6]. Each HRIR is stored in a separate two-channel WAV file in webkit/blink code repository. Since the impulse responses have been normalized in temporal domain to share a common duration (256 samples), the impulse responses have also been concatenated into a single WAV file for faster loading times. The current webkit/blink implementation resorts into the concatenated version using hard-coded sample indices.
The current limitations of such implementation can be summarized in three issues:

1. lack of standardization for the WAV file;
2. no personalization in the HRTF rendering;
3. no support with different databases.

3.2 The SOFA and the Web

SOFA aims at storing data representing HRTFs in a general way, capable of supporting any data measured with microphone arrays and loudspeaker arrays. In particular, SOFA allows a description of a measurement setup with arbitrary geometry, i.e., not limited to special cases like a regular grid, or a constant distance. It provides self-describing data with a consistent definition, i.e., all the required information about the measurement setup must be provided as metadata in the file. It offers flexibility to describe data of multiple conditions (listeners, distances, etc.) in a single file. It supports partial file and network access, and binary files with data compression for efficient storage and transfer.
SOFA aims at representing spatial data in a general way, allowing to store not only HRTFs but also more complex data, e.g., *directional room impulse responses* (DRIRs) measured with a multichannel microphone array excited by a loudspeaker array. In order to simplify the adaption of SOFA for various applications, SOFA supports the so-called conventions: sets of rules for the representation of specific type of data. In this work, two conventions are used: *SimpleFreeFieldHRIR* and *SimpleHeadphoneIR*, which are used to store an HRTF set and headphone IRs, of a listener, respectively.
SimpleFreeFieldHRIR considers the HRTF measurement setup in the free filed with a single excitation source assuming an omnidirectional loudspeaker (Fig. 3) and two receivers at the ears. The measured HRTFs are represented as FIR filters, with a single HRTF set of a listener per file. This convention also describes source position, i.e. azimuth, elevation and radius, and more complex data such as the variation in head tilt or listener position with respect to the center of the measurement setup.

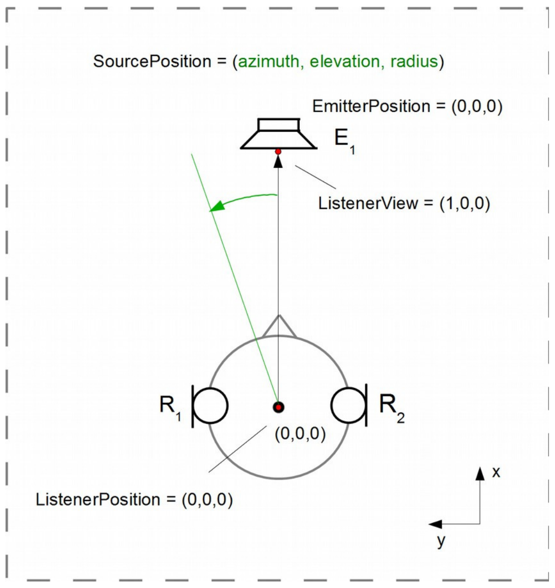
SourcePosition = (azimuth, elevation, radius)
EmitterPosition = (0,0,0)
E_1
ListenerView = (1,0,0)
R_1
R_2
(0,0,0)
ListenerPosition = (0,0,0)
x
y

Figure 3: HRTF measurement setup considered in the SOFA convention SimpleFreeFieldHRIR.

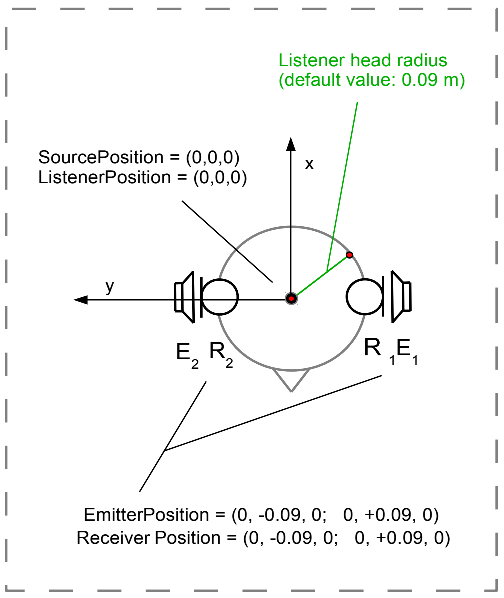
Listener head radius
(default value: 0.09 m)
SourcePosition = (0,0,0)
ListenerPosition = (0,0,0)
x
y
E_2 R_2
R_1 E_1
EmitterPosition = (0, -0.09, 0;   0, +0.09, 0)
Receiver Position = (0, -0.09, 0;   0, +0.09, 0)

Figure 4: HpTF measurement setup considered in the SOFA convention SingleHeadphoneIR.

SimpleHeadphoneIR considers the HpTF measurement setup depicted in Fig. 4. A one-to-one correspondence between emitters and receivers for a single listener is stored in a single file. Multiple measurements (left and right impulse responses) are described as repeated measurements of the same listener with changing in headphone positions or headphone model.
SOFA's numerical container relies on netCDF-4 (Unidata), which is a set of software libraries and data formats supporting the creation, access, and sharing of scientific data. netCDF-4 is widely used in the field of climatology, meteorology,

[6] www.ircam.fr/equipes/salles/listen/

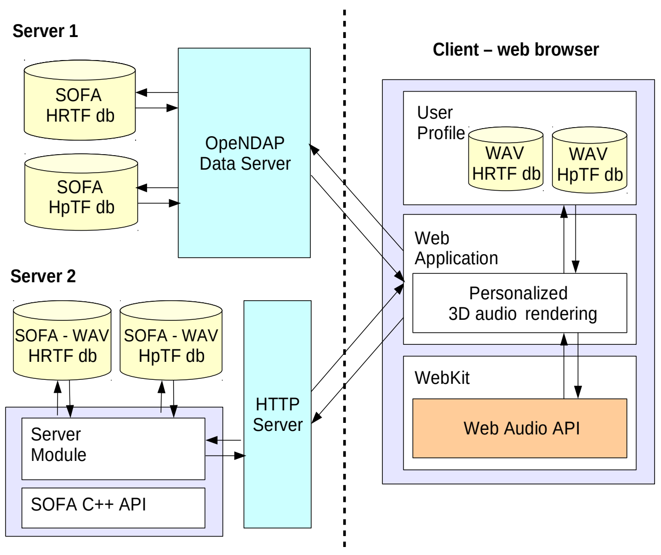

Figure 5: Schematic view of the two proposed web architectures.

oceanography, and geographic information systems. It is based on the HDF5 format, a more basic numerical container, further supported by many institutions worldwide. netCDF offers a structured representation of multidimensional data and metadata. The open-access specifications are freely available and include a complete definition as well as examples of various implementations. Application-programming interfaces are available as pre-compiled libraries for programming languages like C++, Octave, and JAVA.
As for a web application, the libraries providing read access for HDF5 files are currently not a part of a web standard. Thus, SOFA is currently not supported in a browser. Moreover, the solution with JAVA plug-in should be discourage for security reason and uncertainty of future support from the browsers.

Figure 5 schematizes two options to overcome this limitation and serving SOFA files for web: the OPeNDAP[7] server solution and the SOFA server module with WAV conversion.

3.2.1 OPeNDAP

As discussed above, the numerical container of SOFA is based on HDF5. HDF5 is very flexible, and therefore also complex. A C library is provided to hide low level complexities from applications. The library is fairly large in size, and unfortunately insecure, which hinders its adaptation in client side browser code: the source code needs to be carefully analyzed in order to make it trusted, and due to its size, it may require some effort to maintain.

These HDF5 issues make SOFA more suited for server side solution. Thus, the first solution is to employ an OPeNDAP aware server, such as the latter version of OPeNDAP Data Server Hyrax-5. For instance, Microsoft's Azure cloud service has Hyrax-5 integrated, and can therefore serve HDF5-based files, such as SOFA, or parts of the files using OPeNDAP protocol.

3.2.2 Custom SOFA server module

Another option to serve SOFA files for web is to use a conventional HTTP server, and implement a custom SOFA file parser in C++ or C#. This option can be built on top of the HDF5 C library to gain higher semantic level access to SOFA files, e.g., using a managed C# Scientific DataSet library (SDS)[8]. The protocol between client and server is HTTP, which of course is the lingua franca of browser communication and therefore supported ubiquitously. A RESTful protocol is foreseen beneficial.

The server reply to a client HRTF request contains the impulse response itself, and metadata describing elevation, azimuth and distance of the measurement, ITD values, and metadata describing the format of the impulse response (such as sample rate, number of bits per sample etc). One option for encapsulating this data is to use WAV files, which are based on chunk-based *Resource Interchange File Format* (RIFF). Instead of transmitting concatenated impulse responses and resorting to fixed response lengths, the WAV structure could utilize the list-form wavl chunk. A standards compliant form for this is

RIFF { WAVE { fmt wavl { data data data ... } }},

where each data chunk contains a single binaural impulse response. The metadata may be encoded into a newly introduced info chunk, which contains attributes

```
typedef struct info
{
   WORD elevation;
   WORD azimuth;
   WORD distance;
   float itdL;
   float itdR;
}
```

The info chunks can then be appended into the RIFF file in the order of the data chunks. This format enables existing audio editors to be used in editing the wave data. However, if editing is not necessary, the recommended form is to introduce a new 'list' chunk, which contains a sequence of 'hrir' chunks. Each 'hrir' chunk in turn comprises a pair of 'info' and 'data' chunk. The RIFF structure then becomes

RIFF {WAVE {fmt list { hrir hrir hrir ... } }},
with hrir ::= { info data }

In addition to being used as a transfer format, WAV files may also be used to store the fetched HRIRs at client side.

This solution fits well into the current web technology stack. The benefit of using a RESTful protocol and HTTP is obvious: it is both clear and well supported. The benefit of using RIFF based WAV files is twofold: firstly, WAV files are already supported in

Web Audio API. The handling for the proposed new chunk types requires only minor modifications to the trusted browser codebase. Secondly, the chunked nature of RIFF files leaves room for future updates in the spec.

3.3 Implementation

The server side implementation was based on the 2nd solution (Sec. 3.2.2). A custom SOFA file parser implemented on top of SOFA C++ library converts SOFA files to WAV file (with the custom structure described in the previous section). The HRTF and HpTF databases count many impulse responses from the SOFA project[9] and PHOnA archive [20]. The batch conversion is

[7] http://www.hdfgroup.org/projects/opendap/

[8] http://research.microsoft.com/en-us/projects/sds/

[9] http://www.sofaconventions.org/

performed every time a new HRTF/HpTF set is added to the database in the server. Accordingly, new headphone equalization filters are computed and those based on an average of the available measurements are also updated, if necessary.
To access a set of HRTFs for a single subject, the browser issues a GET request of form

http://serverIP/database/subjectName.

To get a specific set of impulse responses for a single elevation, the request may be appended with a tail /elevation, and to access a specific azimuth from the elevation set, tail the request with /azimuth value. For instance, to get all HRTFs for a median plane for subject S001 of the LISTEN database, the browser would issue an *XMLHttpRequest* with URL

http://serverIP/IRCAM/S001/0,

and to get a single binaural HRTF for 90 degree azimuth, the URL would read

http://serverIP/IRCAM/S001/0/90.

More elaborate requests involving several databases, elevations, or azimuths may be handled by inserting the query into the body of the request.

The client side implementation was realized by converting the webkit C++ PannerNode HRTF algorithm (webkit revision R174089, 29/9/2014) into JavaScript by hand. The scripted implementation employs native Web Audio API convolver, delay and gain nodes for improved efficiency. The original C++ implementation is available at [22] and [23], and works as follows.

The webkit HRTF algorithm encapsulates the impulse responses and group delays in four core C++ classes (see pseudocode below for container hierarchy). The *HRTFDatabaseLoader* factory class keeps track of loader instances on a sample rate basis, and loads the impulse response database asynchronously into an *HRTFDatabase* instance. The instance comprises ten *HRTFElevations* distributed from -45 to 90 degrees in 15 degree increments, without interpolation. Each *HRTFElevation* contains two lists of *HRTFKernel* instances, one for the left and another for the right ear. Both lists are divided into 24 raw azimuths angles in 15 degree increments, covering the entire 360 degree space around the horizontal plane. Each raw azimuth interval is further interpolated into eight subintervals, giving rise to 192 distinct *HRTFKernels* per elevation and per ear (i.e., the resolution is less than 2 degrees). Each *HRTFKernel* object encapsulates the measured HRIR, and computes an average group delay for it.

```
AudioContext
- HRTFDatabaseLoader      // one per sample rate
  - HRTFDatabase
    - HRTFElevation[10]   // elevations -45..90°
      - HRTFKernel[24*8]  // azimuths 0..360° (L)
      - HRTFKernel[24*8]  // azimuths 0..360° (R)
        - hrir            // as FFTFrame
        - aveGroupDelay   // float
```

The actual panning algorithm is encoded in three additional C++ classes, cf. Figure 6. Browser audio engine pulls sample buffers periodically from all *PannerNode* instances present in the audio graph (each of which thus models a spatial audio source in 3D space). For each pull, PannerNode first computes its elevation and azimuth in relation to a singleton *AudioListener* (not depicted), and then delegates the computation to an encapsulated *HRTFPanner* instance. *HRTFPanner* fetches the left and right ear *HRTFKernels* according to computed elevation and azimuth values, processes the input with interpolated delay lines for ITD

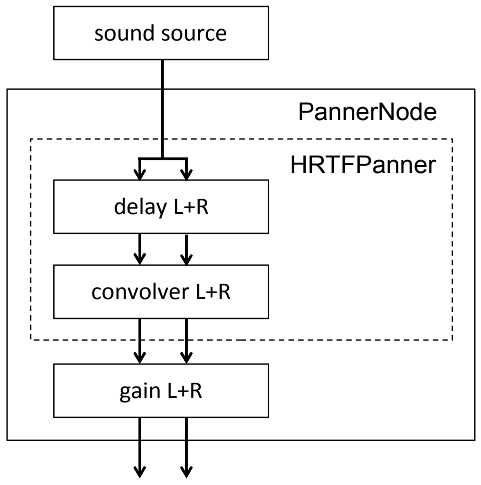

Figure 6: The panning algorithm.

cues, and convolves the result with interpolated left and right ear kernel HRIRs. Additional crossfading is employed for moving sound sources. The produced sample buffers are passed back to the PannerNode instance, which scales the output using distance model and directivity of the sound source, and finally returns the processed buffers for audio engine rendering.

According to the headphone used, the web application queries the server for the equalization filters. Once the server computes the filters, it returns a stereo WAV file that is used in *ConvolverNode* in order to perform a headphone compensation for the left and right channels, respectively.

We conclude from the implementation that enabling custom HRTF rendering in webkit requires only minor modifications to the existing codebase. Firstly, the *HRTFDatabase* needs to be modified to enable networked fetching of impulse responses for individual HRTFs and HpTF equalization. Secondly, the WAV file handling logic needs to be modified to support the proposed chunks. The required modifications are minor when compared to enabling HDF5 support at client side.

4. CONCLUSIONS AND PERSPECTIVES

The proposed architecture supports an individual listening experience within the web browser. Listener can switch between different headphone equalization filters and download his/her HRTFs from the server accordingly by using any of his/her devices. At the moment, publicly available SOFA databases allow to select an existing HRTF set for the spatialization and an existing HpTF set for the headphone equalization.

Once the identity of the listener is known and her/his individual HRTF and HpTF sets are available at the server, the browser incorporates the fetched filters associated with identity information, e.g., anthropometric data, for further usage in web applications of the individualized audio rendering pipeline. WAV format is one potential solution, fully compliant with existing standards. Note that other formats exist supporting web integration, e.g. xml, and json, as well.

If the listener's individual HRTFs are not available, but listener's anthropometric data are known, the web service may perform a customization procedure by exploiting the following solutions:

- anthropometric-guided HRTF selection [19]
- cloud simulation of the whole HRTF set from mesh models/photos of the listener [21].

Considering the web services for listener's individual HpIRs, more difficulties are expected because the available models for the customization of HpTFs are based on average measurements and yield unwanted spectral coloration in the most cases. Development of more sophisticated HpTF models will be important for the future of personalized binaural reproduction.

5. ACKNOWLEDGMENTS

This work was supported by the research project *"PADVA - Personal Auditory Displays for Virtual Acoustics"* (no.CPDA135702) of the University of Padova and by the Austrian Science Fund (FWF, P 24124). We thank Harald Ziegelwanger for providing the figures on HRTF simulations.

6. REFERENCES

[1] M. Morimoto, "The relation between spatial impression and the precedence effect," in *Proc. International Conf. on Auditory Display*, 2002, pp. 297–306.

[2] B. Xie, *Head-related transfer function and virtual auditory display.* Plantatation, FL: J. Ross Publishing, 2013.

[3] L. Savioja, "Modeling Techniques for Virtual Acoustics," Helsinki University of Technology, 1999.

[4] F. L. Wightman, and D. J. Kistler, "Headphone simulation of free-field listening. I: Stimulus synthesis." *J. Acoust Soc Am*, vol. 85, no. 2, pp. 858–867, 1989.

[5] P. Majdak, Y. Iwaya, T. Carpentier, R. Nicol, M. Parmentier, A. Roginska, Y. Suzuki, K. Watanabe, H. Wierstorf, and H. Ziegelwanger, "Spatially Oriented Format for Acoustics: A Data Exchange Format Representing Head-Related Transfer Functions," in *Audio Engineering Society Convention 134*, 2013.

[6] D. R. Begault, A. S. Lee, E. M. Wenzel, and M. R. Anderson, "Direct Comparison of the Impact of Head Tracking, Reverberation, and Individualized Head-Related Transfer Functions on the Spatial Perception of a Virtual Speech Source," in *Audio Engineering Society Convention 108*, 2001.

[7] W. M. Hartmann and A. Wittenberg, "On the externalization of sound images," *J. Acoust. Soc. Am.*, vol. 99, no. 6, pp. 3678–3688, Jun. 1996.

[8] B. Masiero and J. Fels, "Perceptually Robust Headphone Equalization for Binaural Reproduction," in *Audio Engineering Society Convention 130*, 2011.

[9] P. Majdak, P. Balazs, and B. Laback, "Multiple exponential sweep method for fast measurement of head-related transfer functions," *J. Audio Eng Soc*, vol. 55, pp. 623–637, Aug. 2007.

[10] C. P. Brown and R. O. Duda, "A structual model for binaural sound synthesis," *IEEE Trans Speech Audio Lang. Proc*, vol. 6, pp. 476–488, 1998.

[11] S. Spagnol, M. Geronazzo, and F. Avanzini, "On the relation between Pinna Reflection Patterns and Head-Related Transfer Function Features," *IEEE Trans. Audio Speech Lang. Process.*, vol. 21, no. 3, pp. 508–519, 2013.

[12] B. F. Katz, "Boundary element method calculation of individual head-related transfer function. I. Rigid model calculation," *J. Acoust Soc Am*, vol. 110, no. 5 Pt 1, pp. 2440–8, Nov. 2001.

[13] N. A. Gumerov and R. Duraiswami, "A broadband fast multipole accelerated boundary element method for the three dimensional Helmholtz equation," *J. Acoust. Soc. Am.*, vol. 125, no. 1, pp. 191–205, Jan. 2009.

[14] H. Takemoto, P. Mokhtari, H. Kato, R. Nishimura, and K. Iida, "Mechanism for generating peaks and notches of head-related transfer functions in the median plane," *J. Acoust Soc Am*, vol. 132, no. 6, pp. 3832–41, Dec. 2012.

[15] A. Reichinger, P. Majdak, R. Sablatnig, and S. Maierhofer, "Evaluation of Methods for Optical 3-D Scanning of Human Pinnas," in *Proceedings of the 3D Vision Conference*, Seattle, WA, 2013, pp. 390–397.

[16] S. Spagnol, M. Geronazzo, D. Rocchesso, and F. Avanzini, "Synthetic Individual Binaural Audio Delivery by Pinna Image Processing," *Int J. Pervasive Comput. Commun.*, vol. 10, no. 3, pp. 239–254, 2014.

[17] H. Ziegelwanger, P. Majdak, and W. Kreuzer, "Non-uniform sampling of geometry for the numeric simulation of head-related transfer functions," in *Proc. 21st Int. Congress on Sound and Vibration*, Bei-jin, China, 2014, p. Proceedings Number 209.

[18] M. Geronazzo, S. Spagnol, and F. Avanzini, "Mixed Structural Modeling of Head-Related Transfer Functions for Customized Binaural Audio Delivery," in *Proc. 18th Int. Conf. Digital Signal Process. (DSP 2013)*, Santorini, Greece, 2013, pp. 1–8.

[19] M. Geronazzo, S. Spagnol, A. Bedin, and F. Avanzini, "Enhancing Vertical Localization with Image-guided Selection of Non-individual Head-Related Transfer Functions," in *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP 2014)*, Florence, Italy, 2014, pp. 4496–4500.

[20] B. Boren, M. Geronazzo, P. Majdak, and E.Choueiri, "PHOnA: A Public Dataset of Measured Headphone Transfer Functions." *in Audio Engineering Society Convention 137*, 2014.

[21] A. Kärkkäinen, L. Kärkkäinen, and T. Huttunen, "Practical Procedure for Large Scale Personalized Head Related Transfer Function Acquisition," in *Audio Engineering Society Conference: 51st Int. Conf.: Loudspeakers and Headphones*, 2013.

[22] webkit source code for Web Audio API nodes, available online at https://github.com/WebKit/webkit/tree/master/Source/WebCore/Modules/webaudio

[23] webkit platform audio DSP source code, available online at https://github.com/WebKit/webkit/tree/master/Source/WebCore/platform/audio