Spatial audio music player for web

Vojtěch Leischner, Zdeněk Míkovec
Faculty of Electrical Engineering, Czech Technical University in Prague
Karlovo náměstí 13, Praha 2, Czech Republic
leisvoj@fel.cvut.cz

Figure 1: Player different states. From left to right: usage explained, song playing, song selection, information about song author

ABSTRACT

Spatial audio offers new expressive means to music composers. Spatial audio technology is well supported in web browsers for desktop and mobiles platforms alike [**Error! Reference source not found.**]. However, the spatial audio is only marginally represented in web music industry. Most of the online streaming platforms such as Spotify, YouTube or Soundcloud use predominantly stereo format for playback. We propose a new easy to use web-based spatial audio player. We also propose spatial audio encoding using existing audio codecs and discuss the practical implementation. Evaluation of the player was conducted using qualitative user research and by logging measurable data during listening. We aim to democratize the use of spatial audio on the web for non-expert users.

Author Keywords

web audio; spatial audio; music; web design; streaming

ACM Classification Keywords

H. Human-centered computing; H.2 Human computer interaction (HCI); H.2.3 HCI design and evaluation methods

General Terms

Human Factors; Design; Measurement.

DOI: 10.37789/rochi.2021.1.1.19

INTRODUCTION

We need to distinguish between surround audio, ambisonics sound sphere and full featured spatial audio [1,2]. Surround sound uses predefined speaker positions to simulate spatial sound as rendered from a fixed point of view. Both movement and rotation are locked in case of surround audio. This is used mainly in cinemas. In home cinema scenario it is still not widespread [5], but we can see this changing as a streaming platform Netflix supports 5.1 surround sound for some of the streamed movies [3].

Ambisonics sound sphere [**Error! Reference source not found.**] is a representation of sounds coming from different directions recorded from a single point in the space. You can freely rotate inside the sound sphere and audio is rendered dynamically. The rotation can be based on a head movement or other means such as mouse or accelerometer input. It is a hybrid format between audio data format and audio rendering engine.

Full featured spatial audio [**Error! Reference source not found.**] works like our real-world experience as we can rotate and move around in a space and audio changes accordingly. For that we need data for each audio source so we can render audio dynamically based on the listener orientation and position. The listener can freely move between audio sources. Such type of audio rendering is used in computer games and VR.

Our player (see the demo online [29]) has the advantage of full featured spatial audio such as freedom of movement between audio sources. Furthermore, the player is designed for music listening only rather than games and works as a standalone program that does not require any programming knowledge. We also provide a tool [28] for creating the files needed for the spatial playback and tips how to properly record the music. Alternatively, we have also tested audio stem extraction using Neural Networks [23] to separate audio stems from stereo mixdown. Audio stem is a collection of audio sources of similar features, such as all violoncellos in an orchestra. Therefore, even existing mixed down content only available in stereo can still be used with our spatial audio player. The audio stem extraction is not part of our player.

RELATED WORK

The major audio players on the web are services such as Spotify, Apple Music, Gaana, Soundcloud or Deezer [9]. They all stream stereo audio that can be eventually upmixed to 5.1 or other formats but it is just virtualization not a native surround sound and it does not support any movement in the sound sphere. There are also many standalone music players that are using native HTML5 audio capabilities like our player. Most of them do not supports spatial audio or native multi-channel files. However, Trackswitch.js [**Error! Reference source not found.**] audio player supports multiple audio tracks and enables switching between them with radio button controls. Unlike Trackswitch, our player is designed specifically for spatial audio and switching between audio sources is not discrete as in Trackswitch but gradual based on listener distance to the audio source.

There are also web players that support ambisonics formats such as one by Plan8 [10], where the listener can select the normal stereo track that is placed in virtual space that can be rotated around the user. It is different to our player as it only supports one audio source where we support unlimited amount freely placed in space.

Similarly, there are many players for video that support true ambisonics audio such as Hoast library [12], Facebook 360 Studio [13] or even YouTube [14]. Netflix also partially supports 5.1 surround [3]. Unlike Plan8 player [10] these tools are intended mainly for video and they are also limited to rotation of sound sphere only where our player enables absolute movement between individual audio sources.

We should also mention advancements in hardware, particularly Airpods Pro headphones by Apple with integrated Inertial measurement unit that might help to promote the idea of using spatial audio [18] with headphones and offer additional useful sensors to developers.

MUSIC PLAYER APPLICATION

The Player accepts input data in JSON format. Input data includes paths to audio files, color settings, paths to image icons, author information and more. We have integrated our code to work with WordPress CMS [17] as a plugin. This enables us to abstract adding new content with ease through administrator level GUI. WordPress integration is done in PHP and it is completely separated from the player itself. It retrieves information from the database and passes it to the player in JSON format. Manual setup in JSON format without WordPress is also possible. This solution is also convenient for future integration with different databases or CMSs, see Figure 2. Player visualizes the audio sources and listener positions, user than can adjust the position of audio sources or position of the listener, which is immediately reflected in the audio output – see Figure 3.

Front End Graphical User Interface

The player consists of two main areas, see Figure 1, watch video-demo [21] or have a look at live demo online [29]. Top part is the canvas area where the position of audio sources and listener is visualized by circles and ear icon respectively. Circles representing audio sources are animated in real time to indicate the relative volume of the audio source. Player can distinguish between desktop and mobile devices and change controls accordingly.

At the bottom part is the control bar with play / stop button, chosen song title that also acts as songs menu, share button that opens overlay with author specified links and full screen button that spans the player across available window of the web browser.

Figure 2: Data flow from CMS to player

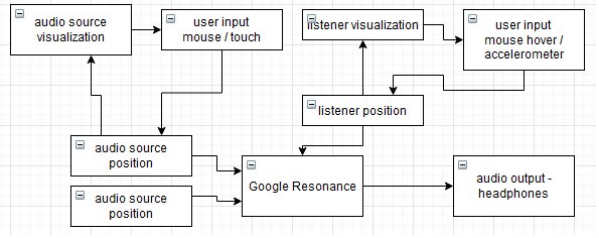

Figure 3: Feedback loop between the player and user

Our goal is to engage the user in active participation during audio listening. User can move individual audio sources represented by circles by dragging them with mouse or touch. Listener position is controlled by the mouse hover or tilting the mobile device to the sides as if the listener was a metal ball on the flat surface that can be balanced. Moving audio sources or listener position immediately changes audio rendering. Thus, the listener creates unique mixdown in real-time.

The visualization part of the website is handled by custom software written mainly in vanilla JavaScript. We are using HTML5 canvas element to draw audio sources and listener icon and regular HTML elements to display the controls.

We have implemented a rudimentary collision detection algorithm to detect user click events to enable dragging individual audio sources. We are using a simple easing algorithm to achieve smooth motion of virtual listener position. This is important because the motion input might not be continuous and if we would drive listener position directly it could produce audio stuttering.

Back End and Audio Engine

Audio rendering is handled by Google Resonance library [**Error! Reference source not found.**] integrated with web SDK. We have chosen Google Resonance audio, because it is specifically designed to support mobile and low performance devices. It creates virtual room, calculate audio signal bounces in the room and model audio reception with accurate ear and skull model using Head related transfer function.

To achieve low latency in audio playback we stream the audio to the user as soon as possible instead of waiting for download of the whole file. We are using HTML5 audio element created dynamically by JavaScript on demand that in turn use prerendered multi-channel audio file. Using multi-channel audio file instead of multiple mono audio sources has added benefit that the audio sources are inherently synchronized. If we would instead use individual mono sources, we could end up with different download speed for each audio buffer and perfect synchronization can prove difficult.

To achieve cross-compatibility we are providing two versions of the file, Ogg Vorbis [19] for Mozilla Firefox and Advanced Audio codec [20] for Chrome, Safari and Microsoft Edge. Both formats support multi-channel encoding. We encode the channels in such a way that every audio source is represented by one audio channel in the file. We are still experimenting with different audio codec settings and compression rate where we are looking to find balance between the file size and quality. We recommend using 64kbps per channel. When the stream is requested by user, we rely on Web Audio API [16] and specifically on Channel Splitter Node [15] to separate individual channels encoded in the audio file. When we extract the audio channels, we pipe them to the Google Resonance [**Error! Reference source not found.**] as individual audio sources – Figure 3.

Recording and exporting Multi Channel Audio Files

To produce multi-channel audio files, we have used freely available open-source program FFmpeg [22]. To further simplify the process, we have created custom made GUI application [28] for the FFmpeg that let the user drag and drop individual audio sources that are combined into one multichannel file and exported as .aac and .ogg files that can be used with the player.

In case of electronic music or combination of electronic instruments and vocals it is easy to obtain individual instruments recordings that can be used to create cleanly separated multi-channel audio. In case of analogue instruments shielding with acrylic plates between musicians can help to some degree. Recording in separate audio booths is better solution but not widely available and expansive.

In the case we cannot get the multi-channel, audio files provided by musician we can opt out for neural network audio stems extractor. It should be stated that properly recording separated audio channels is the preferred option. However, to increase usability of the system we have tested open-source tool Spleeter [23] that enables to extract audio stems from stereo mix. This is useful in case the music was already created and original multi-layer recording was lost. Another use case is for music that is difficult for musicians to record separately because they need to hear each other in real time when performing. Even thought we had only stereo mix-down at our disposal, we were able to get a decent result using free 4 stems pretrained model for bass, vocals, piano, and the rest, see GitHub repository for more [24].

The number of audio channels (audio sources) used is carefully considered. We need to strike a balance between too few and too many. If there would be 256 audio sources, it would disproportionately increase Internet bandwidth. It would also make the interface cluttered, and user could not separate the individual sound sources. As we are limited by screen space especially on mobile devices, we need to provide enough audio sources to keep the complexity but avoid too many to keep the interface well-arranged. 4-6 audio sources seem to work best, 8 is the practical maximum.

EVALUATION

We conducted qualitative user research with the objective to determine if the proposed music player is a viable alternative to stereo format music web players and to evaluate the proposed player. Research was done with 5 participants in their homes where they use their own devices - specification of each device was noted. Participants were briefed about the purpose of the research and instructed to use headphones. First, they were instructed to try the player on the desktop PC and after that on mobile device. Each participant was tasked to go the dedicated website and play a song and listen to it for at least 1 minute. They were also tasked to stop the song, change the song, and explore the options of the player without detailed instructions on how to do that. Besides the structured interview we also log their movement inside the player area. We track the total time duration of the active session, relative distance travelled by listener icon and relative distance travelled while dragging audio sources around. This data acts as a supplement to the structured interview, and we can get insight on how much the user interacted with the player.

Main Research Questions

1. Does a user prefer listening to music using headphones or speakers? What are the use cases for each? What user uses more often?
2. Does the user notice that the audio is spatial, and that the movement causes the changes in the audio rendering?
3. Does the user understand how to control the player?
4. Does the user think that the spatial audio brings new features compared to stereo playback? What does a user find compelling about spatial audio?

5. Would the user listen to his favorite music in spatial format if available? Why?
6. Does user prefer mobile or desktop for online listening music? What are the benefits and disadvantages of using our player on desktop vs mobile?
7. What user likes about the player and what should be changed?

Participants

We have conducted interviews with 5 participants, 4 women and 1 man between 22 and 31 years old. All participants were Europeans. Young age of participants aligned with our target group as we assume that younger people are generally more eager to try new technologies and they are more likely to listen music online [25]. None of the participants were professional musicians. Participants listen to music online in a web browser at least once a week, use both smartphone and pc and own headphones.

Results

All measurements were averaged. Participants spent about 2 minutes listening to the song. They travelled distance with the mouse that equals 55 times the player screen and the distance they dragged individual audio sources equals to 30 times the player screen. Overall, we have observed that participants mainly navigated the listener icon around the audio sources not the other way round. This behavior was same for desktop and mobile. All participants actively explored the movement inside the player audio space.

Headphones vs speakers: Four participants use both headphones and speakers to listen music, one participant mainly uses headphones. Most common use case for headphones was commuting, exercising and generally when you want to avoid bothering other people. Speakers were associated mostly with collaborative listening with friends or at the party, concerts.

Noticing spatial audio rendering: All participants noticed that the audio is rendered differently and that the change is associated with the movement of the listener icon. One participant was wondering if there are other effects involved that are mapped to the mouse movement such as delay, echo, or similar audio effects along with spatial audio. Generally, the spatial property of the sound was well understood.

Player controls: All participants were able to successfully use the player. The issue they spend the time on was figuring out the full screen functionality and recognizing song selection button. We have subsequently improved the player in these areas based on the feedback. All participants understand the mouse hover versus click input on desktop and tilting mechanics on mobile that controls the movement of the listener icon.

Stereo vs spatial: Four participants would consider using the player to listen to their music. One participant felt that the final mix should be done by the musician, fixed and the listener should not get involved. However, rest of the participants enjoyed the experience. One participant said: *"It's definitely a different experience. If the blobs were just playing flat music it would be like skipping through radio stations which isn't exactly a new experience. Spatial audio is more compelling because it is more immersive and helps you to imagine yourself in the digital space."*

Would you use it? Four participants would consider using the player to listen to their music. Another participant was most hopeful for the prospect of using such player for live streams concerts and art installations. Three participants felt that the player is best suited for electronic music. Underlying theme was that the player would be best used with music that was specifically designed for spatial audio. In such a case most, participants would use it repeatedly.

Desktop vs mobile: There was no clear conclusion about either device. Most of the users valued precision and big screen space offered by desktop. On the other hand, participants use the mobile to consume music and find it convenient. We conclude that desktop might be better medium for the player, but we cannot ignore the mobile platform. Therefore, we continue to support both.

Player advantages and drawbacks: Advantage and at the same drawback of the player is required active participation of the user. Player is not well suited for playing music in the background. Users must pay attention to the player and the screen to make use of the spatial rendering.

CONCLUSION

We have created a unique interactive music player for spatial audio listening on web with active user participation using multi-channel audio file streaming. We have verified with users that they recognize the spatial quality of the sound using the player and they would consider it as an alternative to more traditional stereo players.

The developed interface was already used for the release of new album by Aid Kid [26]. We also released commissioned music by collective of composers that ranges from electronic to classical genre. "In terms of sound the audio quality is amazing...It sounds a lot more dramatic in the context of the player, you can feel the dynamic between the instruments in a really unique way I think." Gary Rushton [27], music composer that has used the player.

From the user research we conclude that best content for the player is electronic music. Electronic music is also well suited for creating multi-channel files as we can record individual instruments with ease.

An important thing to note is that the player requires active user participation and therefore it is not suited for background music listening. The main benefit of the player is engaging the user and facilitating part of the music composition process to the listener in user friendly and playful manner. In this regard we have succeeded.

ACKNOWLEDGEMENT

This research has been supported by the project funded by grant no. SGS19/178/OHK3/3T/13.

REFERENCES

1. Gorzel, M., Allen, A., Kelly, I., Kammerl, J., Gungormusler, A., Yeh, H., & Boland, F. (2019, March). Efficient encoding and decoding of binaural sound with resonance audio. In Audio Engineering Society Conference: 2019 AES International Conference on Immersive and Interactive Audio. Audio Engineering Society.
2. Frank, M., Zotter, F., & Sontacchi, A. (2015, March). Producing 3D audio in ambisonics. In Audio Engineering Society Conference: 57th International Conference: The Future of Audio Entertainment Technology–Cinema, Television and the Internet. Audio Engineering Society.
3. Zhang, W., Samarasinghe, P. N., Chen, H., & Abhayapala, T. D. (2017). Surround by sound: A review of spatial audio recording and reproduction. Applied Sciences, 7(5), 532.
4. (2021) 5.1 Surround Sound on Netflix. https://help.netflix.com/en/node/110881/us
5. Frank, M., Zotter, F., & Sontacchi, A. (2015, March). Producing 3D audio in ambisonics. In Audio Engineering Society Conference: 57th International Conference: The Future of Audio Entertainment Technology–Cinema, Television and the Internet. Audio Engineering Society.
6. Vailshery, L.. (2021). Home Theater System/Surround US household penetration 2012-2016. https://www.statista.com/statistics/736072/home-theater-system-surround-us-household-penetration/
7. Rumsey, F. (2013). Spatial audio processing. Journal of the Audio Engineering Society, 61(6), 474-478.
8. (2017). TDG: Personal Audio Headsets Now Used by Three-Fourths of US Adult Broadband Users. https://www.statista.com/statistics/696886/number-of-headphones-owned-in-the-us/
9. Dredge, S.. (2020). How many users do Spotify, Apple Music and streaming services have? https://musically.com/2020/02/19/spotify-apple-how-many-users-big-music-streaming-services/
10. Werner, N., Balke, S., Stöter, F. R., Müller, M., & Edler, B. (2017). trackswitch. js: A versatile web-based audio player for presenting scientific results.
11. (2021) Ambisonics Web Player. https://labs.plan8.se/ambisonics-webplayer/
12. (2021) Higher-Order Ambisonics Streaming Library. https://hoast.iem.at/
13. (2021) Facebook 360 Studio. https://facebook360.fb.com/
14. (2021) Use spatial audio in 360-degree and VR https://support.google.com/youtube/answer/6395969?hl=en
15. (2021) Channel Splitter Node – Web API. https://developer.mozilla.org/en-US/docs/Web/API/ChannelSplitterNode
16. (2021) Web technology for developers. https://developer.mozilla.org/en-US/docs/Web/API/Web_Audio_API
17. (2021) Blog Tool, Publishing Platform, and CMS. https://wordpress.org/
18. Em Lazer Walker. (2020). What is Spatial Audio, Why Does it Matter, and What's Apple's Plan? https://medium.com/@lazerwalker/what-is-spatial-audio-why-does-it-matter-and-whats-apple-s-plan-986f6c662c41
19. Carlacci, A. F. (2002). Ogg Vorbis and MP3 Audio Stream characterization. University of Alberta.
20. Autti, H., & Biström, J. (2004). Mobile Audio-from MP3 to AAC and further. Helsinki University of Technology.
21. Leischner V. (2021), Spatial Audio Player Trick the Ear https://drive.google.com/drive/folders/14tnM0nAEXUWNZDbOjAPCHqmQchIhckRf?usp=sharing
22. (2021) FFmpeg. https://ffmpeg.org/
23. Hennequin, R., Khlif, A., Voituret, F., & Moussallam, M. (2020). Spleeter: a fast and efficient music source separation tool with pre-trained models. *Journal of Open Source Software, 5(50), 2154.*
24. Hennequin, R., Khlif, A., Voituret, F., & Moussallam, M. (2020) Spleeter Github repository. https://github.com/deezer/spleeter
25. Department, P., & 8, J.. (2021). Spotify users by age in the U.S. 2018. https://www.statista.com/statistics/475821/spotify-users-age-usa/
26. Mikula, O. (2021). Aid Kid. https://www.facebook.com/aidkid
27. Rushton, G. (2021). Gary Rushton. https://www.garyrushtonmusic.com/
28. Leischner, V. (2021) Multi-channel audio encoder https://github.com/trackme518/multi_channel_audio_encoder
29. Leischner, V. (2021) Trick The Ear – online player demo https://tricktheear.eu/