Interactive Spatial Audio Rendering on Mobile Devices: A Two-Stage User Interface with Adaptive HRTF Selection and Real-Time Room Acoustics Simulation

Shravan Raghunath[1,*], Kanishk AL[2,*], Sailesh S[3,*], Rishabh Gupta[4,$], Saurav Gupta[5,*], Ramesh R[6,*],
[*]Vellore Institute of Technology Chennai, [$]Samsung R&D Institute, India-Bangalore
[1]shravan.raghunath2022@vitstudent.ac.in, [2]kanishk.al2022@vitstudent.ac.in, [3]sailesh.s2022@vitstudent.ac.in, [4]rishabh.g@samsung.com,
[5]saurav.gupta@vit.ac.in, [6]ramesh.r@vit.ac.in

Abstract

This paper presents an Android-based spatial audio rendering application that provides an immersive audio experience through a two-stage user interface. The first stage focuses on configuring room acoustics and selecting pre-stored HRTFs using a photo of user's face, while the second stage offers an interactive interface for placing sound sources and customizing the environment. The proposed Android application simplifies the process of rendering spatial audio signals with an average latency of 1.27 ms per 20 ms input frame. It maintains a low computational overhead (average CPU usage of 30.6 %), ensuring seamless customized spatial audio processing and rendering over mobile devices. The user interface is designed to be highly intuitive, as validated by subjective evaluations using Mean Opinion Score (MOS) ratings from 15 participants, who reported high user satisfaction and spatial realism (average MOS of 4.72 out of 5). The system supports dynamic listener movement and interactive control over sound source positioning, enhancing sound customization in different acoustic environments. Furthermore, the lightweight implementation with an app size of 104 MB makes it suitable for deployment on a wide range of Android devices, enabling accessible and realistic customization of spatial audio experiences for everyday users.

1 Introduction

The demand for immersive audio is rapidly increasing, driven by gaming and virtual reality applications. Traditional audio systems provide good sound quality but lack the personalization needed for true 3D audio experiences. Humans naturally rely on spatial cues to locate sounds [1], and traditional stereo processing methods cannot re-create the spatial perception at arbitrary positions in 3D space accurately. Early spatial audio solutions often required costly hardware and/or expertise [2], but with advances in mobile processors and spatial audio techniques, smartphones can now support powerful spatial audio applications [3]. A key challenge, however, is designing simple and intuitive interfaces that enable user-friendly spatial audio customization [4].

Spatial audio systems are challenging due to variation in HRTFs across different individuals [2], accurate rendering of room acoustics [5], and the need of real-time low latency processing. Existing approaches often demand expertise and offer limited user control. Although many techniques have been studied in the past [6], creating an intuitive customization for a spatial audio app is still difficult, mainly due to issues such as fast and accurate HRTF selection, room response rendering, virtual object control, occlusion/diffraction modeling, and user movement.

This paper addresses these challenges with a two-stage interface that combines advanced acoustic modeling with intuitive controls. The first stage handles the acoustic environment and HRTF selection, while the second enables interactive sound object movement. This design lets the users experience immersive spatial audio without significant technical expertise, while still benefiting from features like HRTF-based spatialization, realistic room acoustics, real-time occlusion, and interactive object control [2].

The key contributions of the presented work include: (i) computer vision-based head width estimation for personalized HRTF selection, (ii) an automated room acoustics selection system based on user-friendly room size categories, (iii) an interactive virtual environment enabling a high level of user customization, and (iv) advanced acoustic modeling including wall reflections and obstacle occlusion. Our contribution is primarily in the practical implementation, making advanced spatial audio features more accessible and usable on consumer devices.

2 Related Work

Previous research on spatial audio rendering techniques includes several studies related to HRTF measurement and selection of best matching HRTF for individuals from pre-stored databases [7]. The CIPIC database provided a good foundation for research by offering HRTF measurements across subjects [8]. Yet, selecting the best matching HRTFs for each user is an active area of research, with recent work exploring neural fields and diffusion models [9, 10].

Several studies have explored spatial audio rendering on mobile devices, targeting real-time spatialization but often relying on generic HRTFs which can limit the realism for rendered spatial audio [3]. In our review, most previous studies were found to lack systematic description about enabling intuitive usage of spatial audio features on mobile applications, thereby enabling different customizations by non-expert users [6].

Approaches involving computer vision based processing has

979-8-3315-7206-8/25/$31.00 ©2025 IEEE

recently emerged as a promising approach for HRTF selection. Early work by Zotkin et al. used anthropometric measurements [11], while newer studies have proposed applying machine learning for fast, personalized adaptation [9, 10]. Building on this line of work, our photo-based method estimates head size using standard smartphone cameras in a simple and easy-to-use manner.

Room acoustics on mobile devices are often modeled with simplified methods, such as precomputed RIRs and geometric approximations to balance load and realism [12, 13]. We propose to utilize pre-computed, truncated pre-stored RIRs [14], letting users select room sizes (small, medium, large) to adjust realism while keeping latency low.

Mobile User experience (UX) research further stresses intuitive navigation, progressive disclosure, and real-time responsiveness [4]. We utilize these established mobile UX fundamentals for our two-stage interface, making personalized spatial audio more accessible and engaging on mobile devices.

3 System Architecture and Design

The proposed spatial audio rendering application employs a two-stage architecture (Stage 1: Setup, Stage 2: Interactive Control) optimized for CPU efficiency on Android devices, as shown in Fig. 1. Implemented in Kotlin with the Android NDK, the system integrates the user interface, head detection, personalized HRTF processing, room simulation, and spatial audio rendering into a unified pipeline. Separate audio threads are used for playback and processing to ensure real-time responsiveness.

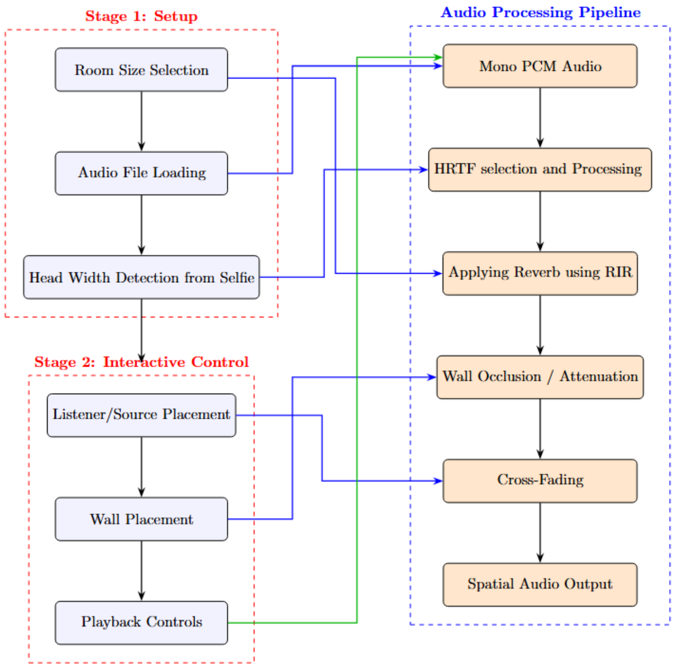


Fig. 1: Block diagram of the proposed spatial audio application with Stage 1 (*Setup*) for input acquisition and Stage 2 (*Interactive Control*) for spatial tuning. The right section shows the audio pipeline converting mono input to spatial audio signals via HRTF, reverb, occlusion, and crossfading.

3.1 Audio Processing Pipeline

The steps for rendering spatial audio, with corresponding inputs and outputs, are as follows:

i. **Mono Input**: Starts with mono PCM audio decoded from .WAV or .MP3 files.

ii. **HRTF Processing**: Applies head-related transfer functions to create left/right ear signals [8].

iii. **Room Simulation**: Adds reverberation effects using room impulse responses (RIRs), simulating different environments [14].

iv. **Occlusion/Attenuation**: Modifies signals to account for walls and obstacles, adjusting sound source levels accordingly.

v. **Cross-Fading**: Smoothly handles source or listener movements, ensuring the signals for different positions transition without artifacts in real time [15].

vi. **Stereo Output**: Generates final stereo PCM audio for playback on headphones or speakers.

3.2 First Stage Interface Design

The first stage interface, as shown in Fig. 2a, is responsible for gathering essential configuration parameters. This interface follows a principle of progressive disclosure, where the user first provides basic inputs (e.g., room type, audio file, and a photo of their face) before moving on to the more detailed spatial configuration settings in the second stage [4].

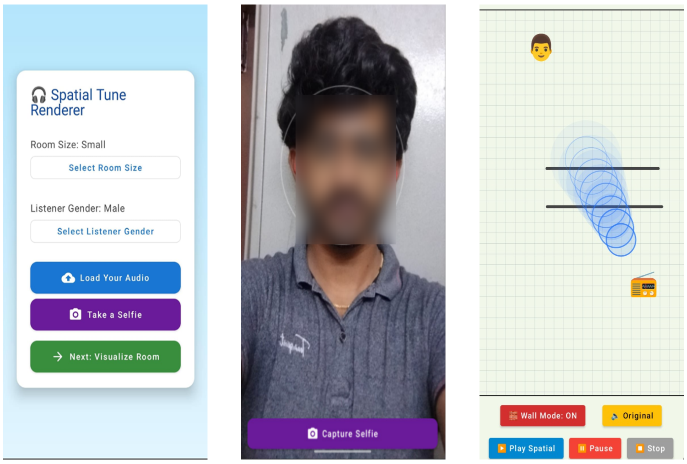
i. Room Size Selection: Users can choose from three room categories. Small rooms (approximately 10 – 30 m^3) represent spaces such as bedrooms or small offices, medium rooms (approximately 30 – 70 m^3) emulate living rooms or small studios, and large rooms (greater than 70 m^3) reflect environments such as auditoriums or concert halls. Each selection loads corresponding pre-computed room impulse responses (RIRs) that model the acoustic characteristics of these spaces [13, 14].

ii. Audio File Management: Users can upload .mp3 or .wav files. The interface provides real-time feedback on file compatibility and allows preview of playback back previews before proceeding.

iii. Photo-Based HRTF Selection: The "Take a Selfie" feature offers a quick, intuitive way to select HRTFs. Users align their head within an on-screen circle (Fig. 2b) to ensure proper positioning and lighting. The image is then processed using facial landmark detection to estimate head width [16, 17], enabling fast HRTF selection without specialized equipment [9].

3.3 Second Stage Interface Design

The second stage interface, as shown in Fig. 2c, helps the user to visualize a room/area and adjust the position of the sound source, listeners, and walls in the interface. This ensures that the spatial audio rendering application is easy to use while ensuring the required functionality. [13].



(a) (b) (c)

Fig. 2: User interface stages: (a) Stage 1 interface, (b) Photo capture interface with face alignment guide for head width estimation to enable HRTF selection, (c) Stage 2 interface.

3.4 Second Stage Interface Design

The second stage interface, as shown in Fig. 2c, helps the user to visualize a room/area and adjust the position of the sound source, listeners, and walls in the interface. This ensures that the spatial audio rendering application is easy to use while ensuring the required functionality. [13].

i. Interactive Grid: The positioning system uses a combination of 2D touch gestures positioning and adjusting the source position accurately in terms of azimuth and elevation.

ii. Object Management: Sound sources appear as a virtual object graphic, such as a radio shown in Fig. 2c, while listeners are represented as characters based on their indicated gender choice (male or female) made in Stage 1. Users can place or drag virtual objects, obstacles, walls, etc., freely within the virtual room environment, with the processing algorithms adjusting the spatial audio signals according to user inputs.

iii. Wall Mode and Acoustic Modeling: When the wall mode is activated, users can draw walls by connecting points on the grid. The spatial audio rendering system automatically computes the spatial audio signal based on wall position/orientation and utilizes the appropriate algorithms to model the dominant sound reflection from the walls [18].

iv. Mono and Spatial Audio: The application allows the user to select and play either the mono audio signal that is pre-stored on the mobile devices or the rendered spatial audio signals to help the user compare the two signals.

4 Technical Implementation

The proposed system focuses on low-latency, real-time spatial audio rendering optimized for mobile devices. It integrates key modules such as HRTF selection, room acoustics simulation, spatial interaction, and occlusion modeling into an efficient audio pipeline. The following subsections describe the core strategies utilized in our spatial audio rendering application.

4.1 Head Width Estimation Algorithm

The head width is estimated by allowing the user to take a photo and computing the head width from that. Instead of relying on external depth sensors, which require specialized equipment, the approach uses relative pixel measurements and a reference scale, which is obtained by directing the user to place their head in a predefined circle drawn in the photo capture UI.

To enable a simple and lightweight implementation suitable for mobile devices, we classify HRTFs into three generalized head sizes (Small, Medium, Large),and the algorithm follows the process outlined below:

i. Face Detection: A photo is captured using the front-facing camera. A face detection API [20] is used to automatically identify and locate the face in the image.

ii. Landmark Detection: The left and right ear landmarks are detected. If both ears are detected, their pixel coordinates are used to calculate a 2D Euclidean distance between them. If detection fails or the photo is unclear, the user is prompted to retake the photo to ensure reliable measurements [17].

iii. Reference Scaling: The photo capture UI shows a circular guide for face alignment. When the face fits this circle, a fixed camera-to-subject distance is assumed, allowing ear-to-ear pixel width to be scaled using a reference(15.5 cm $\approx$ 480 pixels) width calculated from average of ten known subjects.

iv. Width Calculation: The pixel distance between ears is calculated using the linear reference ratio as found in the previous step to estimate head width in centimeters:

$$w_{head} = \frac{d_{eartoear}}{480} \times 15.5 \tag{1}$$

where $d_{eartoear}$ is the pixel distance between the detected left and right ears [16].

Based on the estimated head width, users are assigned an HRTF category [19]:

$$HRTF_{category} = \begin{cases} \text{Small}, & \text{if } w_{head} < 15.0 \text{ cm} \\ \text{Medium}, & \text{if } 15.0 \text{ cm} \leq w_{head} < 18.0 \text{ cm} \\ \text{Large}, & \text{if } w_{head} \geq 18.0 \text{ cm} \end{cases} \tag{2}$$

This approach provides a simple and effective way to select appropriate HRTF from the CIPIC database, without requiring sophisticated hardware or complex models.

4.2 Simulation of Room Acoustics

To simulate reverberation for different rooms, the system employs a simplified method based on stored room impulse responses (RIRs) [14]. Instead of computing acoustic reflections in real time, the application allows the user to select the room response based on three different room sizes: Small, Medium, or Large, and then assigns the selected RIR. Users can dynamically switch between the three different room types in the application.

Each stored RIR aims to model the acoustic characteristics of the room classified based on three different sizes, as mentioned above. The stored RIRs are then convolved with the audio signal obtained after HRTF processing, using efficient time-domain convolution techniques to simulate reverberation [13].

$$\begin{aligned} y_L(t) &= [x(t) * h_{\text{HRIR},L}(t)] * h_{\text{RIR}}(t) \\ y_R(t) &= [x(t) * h_{\text{HRIR},R}(t)] * h_{\text{RIR}}(t) \end{aligned} \tag{3}$$

where $x(t)$ is the dry mono input audio signal selected by the user, $h_{\text{HRIR},L}(t)$ and $h_{\text{HRIR},R}(t)$ are the left and right head-related impulse responses (HRIRs), and $h_{\text{RIR}}(t)$ is the chosen room impulse response (RIR). The resulting signals $y_L(t)$ and $y_R(t)$ represent the final processed spatial audio output signals for left and right ears, respectively

This static-RIR-based approach allows the app to provide immersive room acoustics on mobile devices while reducing the processing latency.

4.3 Dynamic Crossfading

In our proposed application, users can interact with the virtual environment by dragging the source or listener objects to change their positions. Whenever any of the sound objects is moved, the system dynamically updates the spatial audio signals in real time, recalculating the relative direction and distance between source and listener, and applying updated HRTFs to match the new positions [15].

Instead of relying on complex or computationally heavy crossfading algorithms, the application refreshes and renders the new audio output immediately. To avoid artifacts such as glitches or additional latency due to buffers during rapid source or listener movements, the system uses a real-time synchronization clock that keeps position updates aligned with the audio stream, ensuring smooth transitions.This design choice allows for smooth and seamless transitions without introducing additional latency or artifacts, maintaining a consistent 3D audio experience as the user explores different spatial arrangements.

By directly linking user interaction to real-time audio updates, this method ensures that the spatial audio closely follows the user's movements, providing a highly responsive and immersive experience. Additionally, using proposed method instead of traditional cross fading approaches reduces computational load, enabling real-time performance even on resource-constrained mobile devices.

4.4 Wall Occlusion Modeling

The application uses a simplified line-of-sight model to simulate the sound obstruction effects by walls, thereby avoiding the computational cost of detailed ray tracing or diffraction methods. This approach enables lower-complexity modeling suitable for real-time spatial audio rendering on mobile devices [18]. If the direct path between the source and listener intersects any user-defined walls, each crossing is treated as an occlusion, and frequency-dependent attenuation is applied accordingly.

The attenuation is modeled as:

$$A(f) = 10^{-\alpha(f)\cdot n} \tag{4}$$

where $A(f)$ is the attenuation at frequency f, n is the number of wall crossings, and $\alpha(f)$ is a damping factor. The value of $\alpha(f)$ was tuned through controlled listening tests with 10 participants using a Mean Opinion Score (MOS) protocol, where participants rated the perceived realism of obstructed spatial audio signals. The approach is inspired by the Uniform Theory of Diffraction (UTD) [18] and perceptual studies that confirm higher perceptual sensitivity to high-frequency attenuation when direct sound paths are blocked [21]. This model prioritizes real-time performance by assuming straight propagation paths, uniform wall properties, and static listener/source positions. While these assumptions reduce physical accuracy, they achieve improved efficiency for mobile playback.

4.5 Latency Reduction by Truncation of HRIR and RIR

Latency reduction is one of the most critical challenges in real-time spatial audio rendering, especially on mobile platforms. High-resolution head-related impulse response (HRIR) and room impulse response (RIR) datasets can improve the spatial perception of rendered audio; however, the computational complexity of storing and applying these datasets is high, leading to significant processing latency.

In our proposed application, the HRIR and RIR datasets used in this study were truncated to remove the near-zero tail values, thereby reducing convolution operations and computational load while preserving perceptually relevant spatial cues. Using MATLAB, we analyzed impulse responses of HRIRs (at ∼60° azimuth) and RIRs (for a small room and large hall), as shown in Fig. 3 and Fig. 4. The low-energy tail regions, where the signals decayed near zero, were removed. After truncation, the HRIR length was reduced to approximately 128 samples (≈ 2.9 ms at 44.1 kHz), and the RIR length was reduced to approximately 2048 samples (≈ 46 ms at 44.1 kHz). This approach significantly improved computational efficiency for mobile devices while maintaining similar perceptual quality.

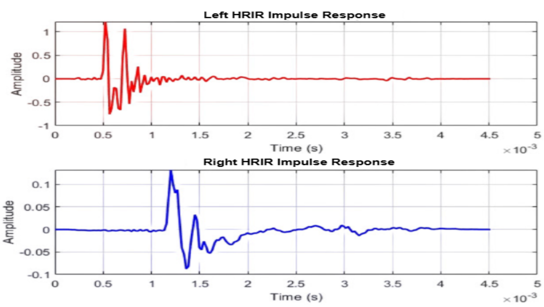


Fig. 3: HRIR after truncation to 3 ms for reducing processing latency.

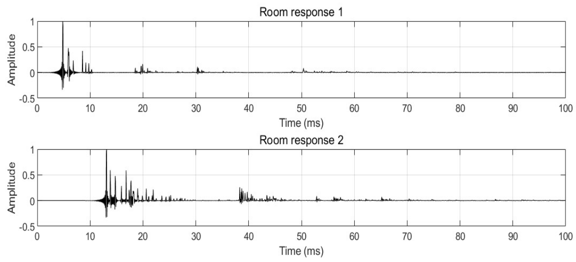


Fig. 4: RIR truncation to 50 ms to reduce processing latency.

5 Performance Metrics

CPU usage was monitored using Android Studio's profiler on a modern high-end Android device (8-core CPU at 2.4 GHz, 8 GB RAM, Android 12). Table 1 lists the performance metrics of the application on this device. Usage was higher during setup and exports, but stabilized to an average of about 30.6% during playback. On mid-range devices, usage is expected to be slightly higher, yet results confirm that the system is lightweight and runs across different mobile devices. These measurements reflect a deliberate trade-off: to ensure low latency and real-time performance, spatial audio processing employs low-complexity methods, improving computational efficiency at the cost of some perceptual accuracy.

For latency analysis, a 5-second audio clip split into 20 ms frames was tested, measuring time for HRTF and RIR processing. The average latency per input frame was 1.268 ms, which was found to be suitable for real-time spatial audio playback during the evaluation.

Overall, these results demonstrate that the proposed application delivers satisfactory real-time performance with efficient CPU use on modern Android mobile devices.

Table 1: Performance Metrics

Metric	Value
Average CPU Usage	30.6%
Application Size (On Disk)	104 MB
Input Frame Size	20 ms
Avg. Latency per Input Frame (HRTF + RIR)	1.268 ms

6 Evaluation and Results

We conducted comprehensive evaluations of our spatial audio rendering application using Mean Opinion Score (MOS) based-ratings. Fifteen participants with normal hearing took part in the evaluation. To ensure consistency, all participants used the same Android device and identical in-ear headphones.

6.1 Subjective Evaluation Protocol and MOS Scores

A user study was conducted to assess audio quality, spatial realism, and usability using Mean Opinion Scores (MOS) on a 5-point scale in steps of 1(1 = poor, 5 = excellent). Non-expert participants evaluated the system under three main criterion:

Criteria 1 — Sound Localization: Participants listened to virtual sound sources placed at various locations while being blindfolded and identified their perceived positions. MOS ratings for this task aimed to examine spatial accuracy of rendered sound sources.

Criteria 2 — Room Acoustics Realism: The same audio was rendered in different virtual room sizes (small, medium, large). Participants rated the perceived realism of the simulated acoustics of different rooms.

Criteria 3 — Interactive Experience: Participants explored the app, simulated acoustics of different rooms, placed and modified walls/obstacles, and rated ease of use and overall satisfaction.

Each trial began with mono playback followed by spatial audio to highlight immersion benefits. In this study, the perceptual evaluation focused more on perceived realism and usability compared to the perceptual accuracy of rendered spatial audio signals.

Table 2: Results for Perceptual Evaluation in terms of Mean Opinion Scores (MOS)

Evaluation Criterion	Mean & Variance of MOS
Spatial Sound Localisation Realism	4.75 ± 0.45
Room Acoustics Realism	4.42 ± 0.49
Ease of Use	4.83 ± 0.39
Overall Satisfaction	4.72 ± 0.36

6.2 Limitations

In our study, despite the encouraging preliminary results from our small scale perceptual evaluation, there are some limitations due to several approximations. The photo-based head width method ignores ear features such as related to pinnae and ear canals, reducing the accuracy of our proposed HRTF personalization algorithm. From previous studies, we know that usage of generic HRTF and RIR databases as well as line of sight occlusion modelling and simplified room acoustics using stored RIRs leads to inaccuracies in the rendering of individualized spatial cues and room acoustics. The design choices made in this study aimed to ensure real-time playback and broad device compatibility, representing a trade-off between computational efficiency and perceptual realism. In our future works, we will focus on advanced techniques for HRTF, RIR and occlusion modelling and analysis of their role in enhancing the perceptual realism and accuracy of rendered spatial audio signals.

7 Conclusion

This paper presents an Android-based spatial audio application that combines real-time simulation with an intuitive, user-friendly interface. The system delivers high-quality, immersive spatial audio while maintaining low latency and smooth interactions on modern devices.

Key contributions of this study are: (1) photo-based HRTF selection, (2) a simple two-stage interface for non-technical users, (3) efficient real-time audio with occlusion and reflections via lightweight methods, (4) promising MOS scores for immersive user experience, and (5) low CPU usage and latency enabling seamless performance on mobiles. The system balances spatial realism, usability, and efficiency, demonstrating that advanced spatial audio rendering applications,can be utilized on mobile devices.

Future works could include using neural field models for more accurate HRTF personalization, together with room acoustics simulations based on ray tracing and more accurate geometric metrics to enhance realism further. [13].

Acknowledgment

We thank Vellore Institute of Technology, Chennai, for their support. We also thank Samsung R&D Institute, India-Bangalore and Mr. C. Omsrinath, and Mr. Abhinandan Udupa and PRISM programme for providing the opportunity and mentorship that played a key role in shaping this project.

References

[1] J. Blauert, *Spatial Hearing: The Psychophysics of Human Sound Localization*. Cambridge, MA: MIT Press, 1997.

[2] W. Zhang, P. N. Samarasinghe, H. Chen, and T. D. Abhayapala, "Surround by sound: A review of spatial audio recording and reproduction," *Appl. Sci.*, vol. 7, no. 5, p. 532, May 2017.

[3] J. A. Belloch, G. Ramos, J. M. Badia, and M. Cobos, "An efficient implementation of parallel parametric HRTF models for binaural sound synthesis in mobile multimedia," *IEEE Access*, vol. 8, pp. 49562–49573, 2020.

[4] C. Wang, L. Chen, L. Zhao, and M. Li, "Exploring the norms for the UX design of intelligent products: a case study," in *Proc. IEEE Tsinghua Int. Design Management Symp. (TIDMS)*, Shenzhen, China, 2013, pp. 158–165.

[5] R. Begault, 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," *J. Audio Eng. Soc.*, vol. 51, no. 10, pp. 904–916, Oct. 2003.

[6] R. Gupta *et al.*, "Augmented/mixed reality audio for hearables: Sensing, control, and rendering," *IEEE Signal Process. Mag.*, vol. 39, no. 3, pp. 63–89, May 2022.

[7] S. Koyama, E. De Sena, P. Samarasinghe, M.R.P.Thomas, and F. Antonacci, "Past, Present, and Future of Spatial Audio and Room Acoustics," IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP), Mar. 2025.

[8] V. R. Algazi, R. O. Duda, D. M. Thompson, and C. Avendano, "The CIPIC HRTF database," in *Proc. IEEE Workshop Appl. Signal Process. Audio Acoust.*, Oct. 2001, pp. 99–102. Available: `https://sofacoustics.org/data/database/cipic/`. Licensed for non-commercial academic use only.

[9] B. Zhi, D. N. Zotkin, and R. Duraiswami, "Towards fast and convenient end-to-end HRTF personalization," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP)*, Singapore, 2022, pp. 441–445

[10] Y. Wang, Y. Zhang, Z. Duan, and M. Bocko, "Predicting global HRTFs from scanned head geometry using deep learning and compact representations," in *Proc. IEEE ICASSP*, 2023.

[11] D. N. Zotkin, R. Duraiswami, and L. S. Davis, "Rendering localized spatial audio in a virtual auditory space," *IEEE Trans. Multimedia*, vol. 6, no. 4, pp. 553–564, Aug. 2004.

[12] T. Wendt, S. P. Par, and S. D. Ewert, "A computationally-efficient and perceptually-plausible algorithm for binaural room impulse response simulation," *J. Audio Eng. Soc.*, vol. 62, no. 11, pp. 748–766, 2014.

[13] L. Savioja and U. P. Svensson, "Overview of geometrical room acoustic modeling techniques," *J. Acoust. Soc. Amer.*, vol. 138, no. 2, pp. 708–730, Aug. 2015.

[14] T. Ko, V. Peddinti, D. Povey, and S. Khudanpur, "A study on data augmentation of reverberant speech for robust speech recognition," in *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2017. Available: `http://www.openslr.org/28`. Licensed under Apache License 2.0.

[15] C. Tsakostas and A. Floros, "Real-Time Spatial Representation of Moving Sound Sources," in 123rd Audio Engineering Society Convention, New York, USA, Oct.2007.

[16] R. A. Ertürk and M. E. Karnaşak, "Anthropometric measurements with 2D images," in *Proc. 7th Int. Conf. on Computer Science and Engineering (UBMK)*, Diyarbakir, Turkey, 2022, pp. 01–06.

[17] Yang, D., Song, C., Zhao, H., Liu, T. (2022). A Facial Size Automatic Measurement and Analysis Technology in Proc. ICPCSEE 2022. Communications in Computer and Information Science, vol 1629.

[18] N. Tsingos, T. Funkhouser, A. Ngan, and I. Carlbom, "Modeling acoustics in virtual environments using the Uniform Theory of Diffraction," in SIGGRAPH, Los Angeles, CA, USA, Aug. 2001, pp. 545–552.

[19] D. Y. N. Zotkin, J. Hwang, R. Duraiswami, and L. S. Davis, "HRTF personalization using anthropometric measurements," in *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, New Paltz, NY, USA, 2003, pp. 157–160.

[20] Sarkar, Sayantan, Vishal M. Patel, and Rama Chellappa. "Deep feature-based face detection on mobile devices." *2016 IEEE International Conference on Identity, Security and Behavior Analysis (ISBA)*. IEEE, 2016.

[21] McSeveney, Steven, et al. "Auditory Occlusion Based on the Human Body in the Direct Sound Path: Measured and Perceivable Effects." 19th International Audio Mostly Conference: Explorations in Sonic Cultures. 2024.