IMPLICIT HRTF MODELING USING TEMPORAL CONVOLUTIONAL NETWORKS

Israel D. Gebru, Dejan Marković, Alexander Richard, Steven Krenn
Gladstone A. Butler, Fernando De la Torre, Yaser Sheikh

Facebook Reality Labs Research, Pittsburgh PA, USA

ABSTRACT

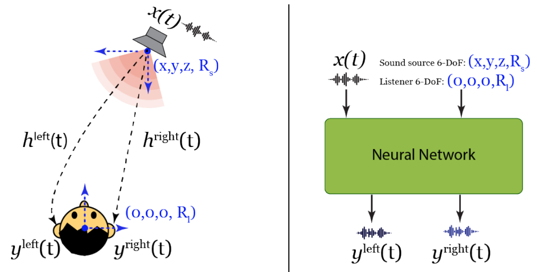
Estimation of accurate head-related transfer functions (HRTFs) is crucial to achieve realistic binaural acoustic experiences. HRTFs depend on source/listener locations and are therefore expensive and cumbersome to measure; traditional approaches require listener-dependent measurements of HRTFs at thousands of distinct spatial directions in an anechoic chamber. In this work, we present a data-driven approach to learn HRTFs implicitly with a neural network that achieves state of the art results compared to traditional approaches but relies on a much simpler data capture that can be performed in arbitrary, non-anechoic rooms. Despite that simpler and less acoustically ideal data capture, our deep learning based approach learns HRTF of high quality. We show in a perceptual study that the produced binaural audio is ranked on par with traditional DSP approaches by humans and illustrate that interaural time differences (ITDs), interaural level differences (ILDs) and spectral clues are accurately estimated.

Index Terms— binaural synthesis, auralization, head-related transfer function, deep learning, spatial audio

1. INTRODUCTION

In many immersive multimedia applications, such as virtual reality, gaming, spatial music, etc., the head related transfer functions (HRTFs) are required to accurately render binaural audio. HRTFs are functions which parameterize the acoustic transfer from a sound source to the ears of a listener. More specifically, they characterize the listener induced changes on the sound field and incorporate cues for sound localization such as interaural time and level differences (ITD, ILD) and spectral cues that arise due to the interaction of pinnae, head and torso with the sound field [1–3]. When a sound signal is filtered with a pair of HRTFs, one corresponding to the path from source to the left ear and the other to the right, and presented through headphones; it gives the listener the impression that the sound source is located in the 3D space [4] (see Figure 1a).

HRTFs can be obtained by means of numerical calculations [5, 6]. To date, however, anechoic measurement is still a common and the most accurate approach, in particular for individual HRTFs of human subjects. Clean, high-quality HRTFs are traditionally measured in an anechoic chamber and require listener-specific recordings at thousands of spatial positions [7, 8]. While these requirements are crucial to enable traditional binaural modeling, data captures are expensive and require highly specialized equipment and capture stages. In this work, we propose a data-driven deep learning approach that circumvents the need for such expensive captures and proves to generate realistic binaural audio with acoustically less treated yet easier to capture data. Moreover, the proposed approach can cope with smoothly changing source/listener positions and does not require physically inaccurate interpolation techniques that are commonly used in traditional approaches.



(a) Traditional synthesis system

(b) Proposed system

Fig. 1: Binaural synthesis systems. **(a)** Traditional binaural synthesis system: input mono signal $\boldsymbol{x}(t)$ is filtered by directional filters $\boldsymbol{h}^{\text{left}}(t)$ and $\boldsymbol{h}^{\text{right}}(t)$ to produce the binaural signals $\boldsymbol{y}^{\text{left}}(t)$ and $\boldsymbol{y}^{\text{right}}(t)$, **(b)** our proposed binaural system using a neural network (NN): takes the mono signal and the source-listener spatial configuration as inputs and outputs binaural signals. $\boldsymbol{R}_s$ and $\boldsymbol{R}_l$ refers the source and the listener's head 3D orientations, respectively.

While the proposed approach is fully data-driven, traditional methods require to deal explicitly with problems due to limited spatial directions. These can be alleviated by forming a continuous, functional representation from sparsely sampled measurements, i.e., expressing the HRTFs mathematically as a continuous function of direction [9]. Methods based on (bi)linear interpolation or cubic spline interpolation [10] that use neighboring HRTFs measurements do not provide sufficiently accurate or high quality HRTFs from sparse measurements, due to the high spatial complexity of the HRTF, especially at high frequencies [11]. Recently more sophisticated methods based on the analysis and decomposition of the entire set of measured HRTFs have been suggested for efficient representation of HRTFs, e.g., using spectral domain [12] and spherical harmonics decomposition [9, 13]. However, these methods are highly constrained on the number and distribution of measurement directions. Moreover, they involve an ill-posed matrix inversion problem that lacks stable solution with respect to data perturbations, and often require arduous regularization [14].

Departing from the conventional signal processing pipeline, where the HRTFs are identified using deconvolution or decorrelation process from anechoic binaural measurement [15] and interpolation needed during synthesis is performed by using one of the aforementioned methods; this paper explores a new machine learning approach. Our motivation is based on the fact that machine learning models can efficiently encode and interpolate data by leveraging domain-specific appearance of signals, and do not impose implicit assumptions (e.g., linearity, minimum phase) constraining conventional HRTFs identification and interpolation methods. Thus, we formulate the problem as a task of estimating masking functions that transform a mono signal into binaural signals. To solve this task, we propose a temporal convolutional neural network (TCN), that depending on the spatial configuration between the source and the listener, predicts the transformation mask –see Figure 1b and Figure 2a. We term the masks as implicit HRTFs since they serve the same purpose as the traditional HRTFs. Application of the mask to the input signal naturally leads to the generation of binaural audio output. At training time, we optimize the network to predict binaural audio such that the HRTFs – for which no ground truth is available – are learned implicitly.

Note that our approach is the first to address the task of learning HRTFs implicitly. There has been some initial work [16–20] where neural networks are used to address mono-to-binaural up-mixing conditioned on video information. These methods, however, treat binauralization as an upmixing task, i.e. the input is the mixed binaural signal, and therefore can not model accurate ITDs and ILDs. By contrast, our model learns to generate binaural sounds depending on source-listener spatial data. Only recently, a WaveNet-based binaural network has been proposed [21] that outperforms traditional HRTF-based techniques; however, the model is domain-specific to speech signals and generally does not provide a solution to estimate general HRTFs on wide-band signals. Our approach, in contrast, learns the signal transformation functions by design and is not constrained to a specific signal domain like speech, as we illustrate in the experiments.

2. PROBLEM FORMULATION

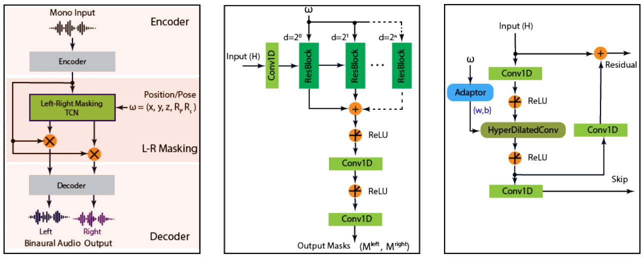
Our main goal is to learn the complex transformation that the sound undergoes as it travels from a position in space and arrives at listener's two ears. We formulate it as a binaural synthesis problem that estimates the transformation of an input mono-source signal $\boldsymbol{x}(t) \in \mathbb{R}^T$ with T samples in time-domain to binaural signals for the left and right ears. The transformation is conditioned on the listener's head orientation given by a rotation vector $\boldsymbol{R}_l$ and the 3D position and orientation of the source $\boldsymbol{P}_s = (x, y, z, \boldsymbol{R}_s)$ expressed in listener's head centered coordinated system – see Figure 1a. We assume that K number of source/listener's 3D positions and orientations are available for the duration of $\boldsymbol{x}(t)$, i.e $1 \leq K \leq T$. The binaural signals for the left- and right ears are given as:

$$(\hat{\boldsymbol{y}}^{\text{left}}(t), \hat{\boldsymbol{y}}^{\text{right}}(t)) = f(x(t), \boldsymbol{\omega}) \qquad (1)$$

where f is a function parameterized by a neural network and $\boldsymbol{\omega} = (\boldsymbol{R}_l^k, \boldsymbol{P}_s^k, \boldsymbol{R}_s^k)_{k=1}^K$ are network conditioning inputs, and are referred to as the direction code. Note that, traditional HRTF representation ignores the orientation of the source since the sound source is assumed to be an ideal source (i.e. omni-directional or point source). In this work, we take into account the orientation of the source since the function $f(.)$, that we are trying to model, is learnt in a supervised manner using data collected from real audio sources (e.g., a human speaker, loudspeaker).

3. PROPOSED MODEL

The block diagram of the proposed neural network model is shown in Figure 2. The architecture is inspired by traditionally binaural synthesis with HRTF filtering; that is, direction dependent filters for the left- and right ears are applied as multiplicative *masks* onto the input mono signal in frequency domain to produce the binaural signals. Our model architecture consists of three processing modules as shown in Figure 2a: an encoder mapping the input signal into a learned frequency space, a left-right masking temporal convolutional network (TCN), and a decoder mapping the transformed signal back into the wave domain. We describe the details of each module in this section.



(a) Block diagram of the proposed network

(b) The Masking TCN Block

(c) The ResBlock

Fig. 2: An overview of the proposed network architecture.

Encoder. The encoder module in Figure 2a transforms short segments of the input mono waveform into their corresponding representations in an intermediate feature space suitable for binaural synthesis. It also generates a residual connection which facilitates the reconstruction of binaural signals by the decoder. The encoder is implemented as a one layer 1-D convolutional neural network followed by a PReLU as a non-linear activation function. We initialized the kernel weights of the convolutional layer with handcrafted representation derived from the bases of the discrete Fourier transform. The weights are further optimized with an end-to-end training paradigm. The particular initialization scheme we used allows easy signal analysis and re-synthesis, and helps the model to preserve network capacity. We found those learned frequency embeddings to yield better results than a fixed discrete Fourier transform in initial experiments. The encoder can also be considered as trainable Short-Time Fourier Transform (STFT) layer [22].

Left-Right Masking TCN. The left-right masking module, shown in the middle of Figure 2a, consists of a temporal convolutional network (TCN) block [23]. It takes the encoder output $\mathbf{H}$ and the direction code $\boldsymbol{\omega}$ as inputs and generate multiplicative masks $\mathbf{M}$ for the left and right channels:

$$(\mathbf{M}^{\text{left}}, \mathbf{M}^{\text{right}}) = \text{TCN}(\mathbf{H}, \boldsymbol{\omega}). \qquad (2)$$

As shown in Figure 2b, the TCN module begins with a linear 1x1 convolutional layer that serves as a bottleneck. This layer determines the number of channels in the subsequent blocks. Figure 2c shows the design of residual blocks. The residual block is composed of a 1x1 convolutional layer for channel mixing, followed by a hyper-convolutional layer with increasing dilation factors, shown as *HyperDilatedConv* in Figure 2c. The dilation factors increase exponentially to ensure a sufficiently large temporal context window to model the long-range dependencies. The skip-connection and residual path designs follows [24, 25]. The residual path of a block serves as the input to the next block, and the skip-connection paths for all blocks are summed up and used as output masks.

Furthermore, a common way to condition a neural network is to add or concatenate some representations before feeding them into its input layers. However, we observed in early experiments that nuanced source/listener position dependent signals are not well modeled by such standard techniques. To adapt the output masks based on the geometric relation between source and listener, we use a hyper-convolution layer similar to what is proposed in [26]. We condition the weights of the network on the source/listener positions and orientations. These direction-dependent weights and biases are obtained from the adaptor network that takes the direction code $\boldsymbol{\omega}$ as an input and outputs the weights and bias which are then used in the temporal convolutional layers. Hence the generated weights and biases contain information about the geometric relation between source and listener.

Decoder. The decoder module transforms the modified (masked) learned frequency embeddings back to waveform domain with a 1-D transposed convolution,

$$\hat{\boldsymbol{y}}^c(t) = \text{TransposedConv}(\mathbf{H} \odot \mathbf{M}^c) \quad c \in \{\text{left}, \text{right}\}, \qquad (3)$$

where $\odot$ denotes element-wise multiplication, and $\mathbf{M}^c$ is the estimated mask. Our intuition behind masking is that when the masks are applied to the encoder output, they cause the input mono signal to undergo the complex transformations that make up the binaural signals, e.g., geometric configuration dependent amplitude scaling and phase shifts. We argue that the masks embed ITDs, ILDs and spectral cues.

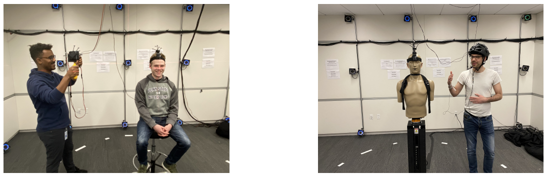
Similar to the encoder, we initialize the kernel weights with Fourier bases and further finetune them together with the other modules. Note that, the decoder module is shared between the left and right channels.

4. DATASET AND IMPLEMENTATION

Dataset. Since there is no publicly available binaural audio dataset with the 6-DoF data as described in Section 2, we have collected a novel dataset with binaural audio recordings from a KEMAR mannequin and 8 human subjects with 6-DoF tracking information to train, validate, and test the proposed model. With the setup shown in Figure 3a, we record binaural audio data as subjects listened to a pink-noise sound playing on a loudspeaker. Subjects sat on a saddle stool in an acoustically treated room and another person with a small loudspeaker[1] walks around the subjects to cover a sufficient amount of spatial locations. The subjects wore a B&K 4101B binaural microphone in the ears and a headband with reflective markers for head-pose tracking. Reflective markers were also placed on the loudspeaker for tracking. The subjects were allowed to rotate and move their heads during the recording. The signal recorded on a microphone that was attached to the loudspeaker is used as the input mono signal for model training. We collected a total of 2.6 hours of audio data (20 minutes from each subject). The audio data is recorded at 48kHz sampling rate and rigid body tracking data is collected at 120fps via motion capture software, Motive. Linear-Time-Codes (LTC) are used to synchronize the audio recordings with tracked source/listener positions. Furthermore, with the setup shown in Figure 3b, we collected additional speech data on KEMAR mannequin while a person speaks and walks around for system validation purposes. We collected 2 hours of data from 8 people speaking to KEMAR. The signal recorded by a lav microphone attached to the side of the face and pointing towards the person's mouth is used as the input mono sound.

Note that this capture setup is much simpler than traditional binaural data captures as neither an anechoic chamber nor a precisely placed speaker array are required. While traditional models require almost ideal HRTF measurements for accurate modeling, our proposed network – as many deep learning, data-driven approaches – can cope with this simpler yet acoustically less ideal data and still produce realistic binaural audio.

Implementation. We outline the implementation details that differ from the common implementation and further refer the interested reader to the code[2] for a full account of them. For the encoder/decoder modules, we set the kernel size, stride with values 40 and 10 respectively. The number of output channels for the encoder is set to 512. We applied ℓ_2-normalization to the filter coefficients in the encoder and decoder modules before computing the convolution to avoid the filters learn scaling factor, which is taken care of by the masking TCN module. Furthermore, we observe that increasing the number of channels, i.e. the number of basis signals, in the encoder/decoder increases the overcompleteness of the basis signals and improves the performance. In our experiments, we use a small number of channels as a trade-off between performance and model size. The masking TCN module consists of three sequential blocks. Each block is a stack of eight layers of ResBlock. The first Conv1D layer reduces the number of channels to 128, and from that we kept the same number of channels in the subsequent blocks. The kernel size in the *HyperDilatedConv* layer is set to 4, and the dilation size is doubled after each layer. The last Conv1D layer increased the number of channels to match twice of the encoder output. The adaptor block is a simple Conv1D layer with a ReLU non-linearity.



(a) Recording setup for training

(b) Recording setup for testing

Fig. 3: Dataset recording setups.

Loss Function. To train our model, we minimize the multi-scale Short-Time Fourier Transform (STFT) loss [27], which has been commonly used to replace point-wise losses on the raw waveforms. Let L_i define a single STFT complex spectrogram l_1 loss with a given FFT size i. The total loss is then the sum of all the spectral losses for the left and right channels $L_{\text{total}} = \sum_i L_i^{(\text{left})} + \sum_i L_i^{(\text{right})}$. We use FFT sizes (2048, 1024, 512, 256), and the neighboring frames in the STFT overlap by 75%.

Training. We train the model using Adam with a batch size of 196. The initial learning rate is 0.0001 with $\beta_1 = 0.9$ and $\beta_2 = 0.999$. We anneal the learning rate by a factor of two if five epochs have passed without improvement in the validation set. We train with a frame size of 8000 audio samples at 48kHz with 50% overlap, 20 samples for direction code.

5. EXPERIMENTS AND DISCUSSIONS

We use our recorded pink-noise binaural dataset for training (2.1 hours of data for training, 0.5 hours for validation). For testing, we use the speech recordings collected on KEMAR. Moreover, to study our model performance on non-speech signals, we added generic sounds such as music, car horn, truck engine, dog barking.

We compare binaural audio synthesized by the proposed model with the ground-truth binaural recordings (if available), and with a DSP approach using HRTF measurements from [28]. To analyze our system, we carry out quantitative and perceptual evaluations. For the perceptual evaluation, we asked 34 people to listen and rate a 15-seconds long audio clips with a mean opinion score (MOS) from 1 to 5. We consider three aspects: naturalness of the signal, spatialization quality, and similarity of the synthesis binaural audio to the actual binaural recording. Naturalness aims to measure the amount of artifacts or distortion that is present in the generated binaural signals. Spatialization measures how well the path of the virtual sound source matches the path rendered in the binaural audio. We provide the participant a video of the sound source path accompanying the

[1] We used MINX MIN 12 speaker for its small size.

[2] Dataset and code will be publicly available

binaural audio. Similarity aims to measure how well the generated binaural signals resembles the binaural recording during playback. Three audio clips are presented to participants in the Naturalness assessment parts (one per system), six audio examples in the spatialization part (two per system), and three recorded/synthesized pairs of binaural audio for similarity evaluation (one per system). The audio clips were randomly selected from a pool of 200 audio clips containing speech and generic sounds samples. As objective metrics, we used the short-time objective intelligibility (STOI) [29] and the Perceptual evaluation of speech quality (PESQ) [30] metrics. STOI prvoides a score from 0 to 100 for speech intelligibility; PESQ provides a voice quality score similar to the mean opinion score given by a human listener.

In Table 1, we present the quantitative and perceptual evaluation results. Our proposed approach performs comparable to the traditional DSP model and differences are all within a statistically insignificant range. Remarkably, our model gets higher scores on speech signals for spatialization and similarity metrics compared to the DSP approach. Recall that our model is trained on pink noise rather than speech signals, so speech spatialization is out-of-domain data that our deep network still handles well. The performance of our method is slightly degraded on non-speech signals for naturalness and spatialization, however, the perceived sound quality and spatial impression are within a small margin to the DSP baseline. The quantitative evaluation (STOI and PESQ) confirms the good performance of our model. Even if our model applies complex non-linear modifications, it does not add artifacts that significantly distort the binaural speech signal. The main advantage of our method is that one can use wide-band signals for training and still accurately synthesize binaural audio on other, out-of-domain signals types. Note that in Table 1, our method proves to spatialize speech particularly well, compared to other sounds including traffic and music. Also note that all subjective scores are below a 5 because participants listened to non-personalized binaural audio on their personal, potentially non-equalized headphones [31].

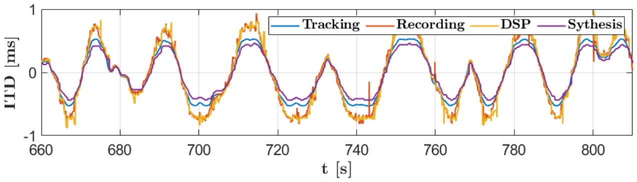


Fig. 4: ITDs obtained from the tracking data (free-field propagation) compared to estimates from recorded and synthesized signals. The synthesized signal exhibits smooth IDTs that align closely with the analytically computed expected IDTs from the tracked source/listener positions.

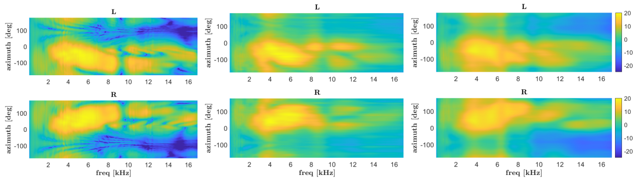


Fig. 5: HRTF magnitude on horizontal plane. **Left**: measured in anechoic chamber, **Middle**: estimated from binaural recordings, **Right**: estimated from signals synthesized using our model.

Table 1: Percpetual scores and quantitative metrics on binaural synthesis. For all measures, higher is better.

	Perceptual Study				Quantitative Eval	
		Spatialization				
	Naturalness	Speech	Other	Similarity	STOI (%)	PESQ
DSP	4.06 ± 0.95	3.71 ± 1.02	4.13 ± 1.13	2.79 ± 1.17	99.9	3.1
Ours	3.79 ± 0.88	3.97 ± 0.88	3.78 ± 1.12	3.18 ± 1.10	98.9	3.6
Recordings	4.19 ± 1.00	4.32 ± 0.81	-	4.77 ± 0.37		

Moreover, as a means to validate the binaural cues learned by our model, we evaluate how well the ITD and ILD in the synthesized binaural audio match the recorded binaural audio signals. We use the binaural recordings from KEMAR collected using the setup shown in Figure 3 for evaluation. Figure 4 shows the results on ITD estimation from a test sequence. Our model is able to produce smooth ITDs over time that closely match the recorded signal's IDTs. Note that the synthsized binaural signals also have IDTs close to the ones that are expected as the ideal, analytical solution based on the tracked source/listener positions. Analyzing the ILDs, we extract the HRTF from the synthesised binaural audio based on the steps describe in Section 5.1. Figure 5 shows the extracted HRTFs magnitude spectra. It can be seen that our model is able to attenuate certain frequencies and boost others with a pattern that closely match anechoic HRTF spectra. Also note that some of the notch are missing in the estimated HRTF spectra, however this is expected since the binaural recordings are done in non-anechoic/non-quiet environment.

5.1. HRTF Extraction

HRTFs extraction from a continuously moving sound source is a not a straightforward process. The steps to get the HRTFs shown in Figure 5 are described here. Let $X(n, f)$ be the STFT of the input signal $x(t)$ and $Y^c(n, f)$ the STFT of the output signals $y^c(t)$, $c \in \{\text{left}, \text{right}\}$, with n and f denoting, respectively, the time frame and frequency band. Let $\phi(n)$, $\theta(n)$, $r(n)$ indicate the tracking data, i.e. the azimuth, elevation and range of the source with respect to the listener at the given frame n. The magnitude of the HRTF is estimated as:

$$|H^c_{eq}(\phi_i, \theta_i, f)| = \frac{G_{ref}(f)}{\sqrt{\frac{1}{M}\sum_{j=1}^{M} |H^c(\phi_j, \theta_j, f)|^2}} |H^c(\phi_i, \theta_i, f)| \,, \quad (4a)$$

$$|H^c(\phi_i, \theta_i, f)| = \sqrt{\frac{1}{N_i} \sum_{n_i \in B_i} \frac{|Y^c(n_i, f)|^2 r(n_i)^2}{|X(n_i, f)|^2}} \,, \quad (4b)$$

$$\begin{aligned} B_i = \{n : |(\phi(n) - \phi_i| < \Delta\phi, |\theta(n) - \theta_i| < \Delta\theta, \quad (4c) \\ r(n) > r_{min}, |X(n, f)|^2 > P_{min}\} \end{aligned}$$

where, given the estimation grid $(\phi_i, \theta_i)_{i=1}^{M}$, the direction-dependent spectral modifications (4b) are estimated by deconvolution process and temporal averaging over all time frames corresponding to the given grid direction (4c). Then, these estimates are refined by the diffuse-field equalization (4a), whose purpose is to remove the influence of the measurement equipment and the environment. Finally, $G_{ref}(f)$ is an optional direction-independent equalization filter that for our results is a diffuse-field average of a reference HRTF.

6. CONCLUSIONS

In this work, we explore the use of neural networks to implicitly learn continuous transfer functions for binaural synthesis. We proposed a temporal convolutional network that estimates the mono-to-binaural transformation function and generates binaural audio from input mono audio based on the source/listener's 6-DoF information. We assess the performance of the network in quantitative and perceptual evaluations. We also demonstrate that our network is able to reproduce the HRTFs, as well as capture ITDs and ILDs binaural cues. Future extensions of this work include personalized implicit HRTFs, in particular using a person's head and pinnae photometric information as additional conditioning parameter.

7. REFERENCES

[1] D. Wright, J. H Hebrank, and B. Wilson, “Pinna reflections as cues for localization,” *The Journal of the Acoustical Society of America*, vol. 56, no. 3, pp. 957–962, 1974.

[2] F. Asano, Y. Suzuki, and T. Sone, “Role of spectral cues in median plane localization,” *The Journal of the Acoustical Society of America*, vol. 88, no. 1, pp. 159–168, 1990.

[3] C. I Cheng and G. H Wakefield, “Introduction to head-related transfer functions (hrtfs): Representations of hrtfs in time, frequency, and space,” *Journal of the Audio Engineering Society*, vol. 49, no. 4, pp. 231–249, 2001.

[4] C. Hendrix and W. Barfield, “The sense of presence within auditory virtual environments,” *Presence: Teleoperators and Virtual Environments*, vol. 5, no. 3, pp. 290–301, 1996.

[5] Brian F. G. Katz, “Boundary element method calculation of individual head-related transfer function,” *The Journal of the Acoustical Society of America*, vol. 110, no. 5, 2001.

[6] T. Xiao and Q. Huo Liu, “Finite difference computation of head-related transfer function for human hearing,” *The Journal of the Acoustical Society of America*, vol. 113, no. 5, 2003.

[7] V R. Algazi, R. O Duda, D. M Thompson, and C. Avendano, “The cipic hrtf database,” in *IEEE Workshop on the Applications of Signal Processing to Audio and Acoustics (WASPAA)*. IEEE, 2001, pp. 99–102.

[8] S. Li and J. Peissig, “Measurement of head-related transfer functions: A review,” *Applied Sciences*, vol. 10, no. 14, pp. 5014, 2020.

[9] M. J Evans, J. AS Angus, and A. I Tew, “Analyzing head-related transfer function measurements using surface spherical harmonics,” *The Journal of the Acoustical Society of America*, vol. 104, no. 4, 1998.

[10] T. Nishino, S. Kajita, K. Takeda, and F. Itakura, “Interpolating head related transfer functions in the median plane,” in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*. IEEE, 1999, pp. 167–170.

[11] Z. Ben-Hur, D. L Alon, R. Mehra, and B. Rafaely, “Efficient representation and sparse sampling of head-related transfer functions using phase-correction based on ear alignment,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 12, pp. 2249–2262, 2019.

[12] M. Matsumoto, S. Yamanaka, M. Toyama, and H. Nomura, “Effect of arrival time correction on the accuracy of binaural impulse response interpolation–interpolation methods of binaural response,” *Journal of the Audio Engineering Society*, vol. 52, no. 1/2, pp. 56–61, 2004.

[13] R. Duraiswaini, D. N Zotkin, and N. A Gumerov, “Interpolation and range extrapolation of hrtfs [head related transfer functions],” in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*. IEEE, 2004, vol. 4, pp. iv–iv.

[14] D. N Zotkin, R. Duraiswami, and N. A Gumerov, “Regularized hrtf fitting using spherical harmonics,” in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*. IEEE, 2009, pp. 257–260.

[15] A. Novak, L. Simon, F. Kadlec, and P. Lotton, “Nonlinear system identification using exponential swept-sine signal,” *IEEE Transactions on Instrumentation and Measurement*, vol. 59, no. 8, pp. 2220–2229, 2009.

[16] P. Morgado, N. Nvasconcelos, T. Langlois, and O. Wang, “Self-supervised generation of spatial audio for 360 video,” in *Advances in Neural Information Processing Systems*, 2018.

[17] R. Gao and K. Grauman, “2.5d visual sound,” in *IEEE Conf. on Computer Vision and Pattern Recognition*, 2019.

[18] YD. Lu, HY. Lee, HY. Tseng, and MH. Yang, “Self-supervised audio spatialization with correspondence classifier,” in *IEEE International Conference on Image Processing*, 2019.

[19] K. Yang, B. Russell, and J. Salamon, “Telling left from right: Learning spatial correspondence of sight and sound,” in *IEEE Conf. on Computer Vision and Pattern Recognition*, 2020.

[20] H. Zhou, X. Xu, D. Lin, X. Wang, and Z. Liu, “Sep-stereo: Visually guided stereophonic audio generation by associating source separation,” in *European Conf. on Computer Vision*, 2020.

[21] A. Richard, D. Markovic, I. D Gebru, S. Krenn, G. Butler, F. de la Torre, and Y. Sheikh, “Neural synthesis of binaural speech,” in *International Conference on Learning Representations*, 2021.

[22] M. Pariente, S. Cornell, A. Deleforge, and E. Vincent, “Filterbank design for end-to-end speech separation,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 6364–6368.

[23] C. Lea, M. D Flynn, R. Vidal, A. Reiter, and G. D Hager, “Temporal convolutional networks for action segmentation and detection,” in *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 156–165.

[24] A. v. d Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, “Wavenet: A generative model for raw audio,” 2016.

[25] Yi Luo and N. Mesgarani, “Conv-tasnet: Surpassing ideal time–frequency magnitude masking for speech separation,” *IEEE/ACM transactions on audio, speech, and language processing*, vol. 27, no. 8, pp. 1256–1266, 2019.

[26] D. Ha, A. Dai, and Q. Le, “Hypernetworks,” *arXiv preprint arXiv:1609.09106*, 2016.

[27] R. Yamamoto, E. Song, and JM. Kim, “Parallel wavegan: A fast waveform generation model based on generative adversarial networks with multi-resolution spectrogram,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 6199–6203.

[28] Z. Ben-Hur, D. Alon, P. W. Robinson, and R. Mehra, “Localization of virtual sounds in dynamic listening using sparse hrtfs,” in *Audio Engineering Society Conference: 2020 AES International Conference on Audio for Virtual and Augmented Reality*. Audio Engineering Society, 2020.

[29] C. H. Taal, R. C. Hendriks, R. Heusdens, and J. Jensen, “An algorithm for intelligibility prediction of time–frequency weighted noisy speech,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 7, pp. 2125–2136, 2011.

[30] A. W Rix, J. G Beerends, M. P Hollier, and A. P Hekstra, “Perceptual evaluation of speech quality (pesq)-a new method for speech quality assessment of telephone networks and codecs,” in *IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings*. IEEE, 2001, vol. 2, pp. 749–752.

[31] B. Masiero and J. Fels, “Perceptually robust headphone equalization for binaural reproduction,” in *Audio Engineering Society Convention 130*. Audio Engineering Society, 2011.