# Deep HRTF Encoding & Interpolation : Exploring Spatial Correlations using Convolutional Neural Networks

**Devansh Zurale**
UC San Diego
dzurale@ucsd.edu

**Shahrokh Yadegari**
UC San Diego
sdy@ucsd.edu

**Shlomo Dubnov**
UC San Diego
sdubnov@ucsd.edu

## ABSTRACT

With the advancement in Deep Learning technologies, computers today are able to achieve unimaginable success in several domains involving images and audio. One such area in 3D audio where the applications of deep learning can be promising is in binaural sound localization for headphones, which requires individualized and accurate representations of the filtering effects of the anthropometric measurements of a listening body. Such filters often are stored as a set of Head Related Impulse Responses (HRIRs) or in their frequency domain representations, Head Related Transfer Functions (HRTFs), for specific individuals. A challenge in applying deep learning networks in this area is the lack of availability of vast numbers of complete and accurate HRTF datasets, which is known to cause networks to easily over-fit to the training data. As opposed to images, where the correlations between pixels are more statistical, the correlations that HRTFs share in space are expected to be more a function of the body and pinna reflections. We hypothesize that these spatial correlations between the elements of an HRTF set could be learned using Deep Convolutional Neural Networks (DCNNs). In this work, we first present a CNN-based auto-encoding strategy for HRTF encoding and then we use the learned auto-encoder to provide an alternate solution for the interpolation of HRTFs from a sparse distribution of HRTFs in space. We thereby conclude that DCNNs are capable of achieving results that are comparable to other non deep learning based approaches, in spite of using only a few tens of data points.

## 1. INTRODUCTION

With the increasing demand for Virtual Reality (VR) technology, there is an increasing need to accurately model sound localization. When a sound source originates from a particular point in space around the head, it undergoes different sets of reflections off of body parts and the pinna before reaching each ear. This difference in reflections, along with the difference in times of arrival and the amplitudes of arrival of the sound source at each ear is what is understood to give humans the perception of direction of the sound source. In order to then add direction to a sound source being played through headphones, one needs to filter the sound source with the transfer functions between the desired points of origination of the sound source and the points at which the sound signal enters each ear. These transfer functions, known as the Head Related Transfer Functions (HRTFs), or as their time domain representations, the Head Related Impulse Responses (HRIRs), are unique for every point in space, for each ear and for every individual.

To allow for the sound source to be perceived as originating from every direction in space, every individual would need to measure and store the HRTFs for each ear, ideally for every point in space around the head, or practically at least sampled at several hundred points. However, having to measure as many HRTFs for an individual is a time consuming and tedious process that requires very sophisticated and expensive equipment, making it impractical for consumers to record their own sets of HRTFs. One way then to be able to give binaural spatial audio experience to the listeners is to either provide a generic non-individual HRTF, or present the users with several non-individual HRTFs to pick from, the one that works best. Such non-individual HRTFs however, are known to introduce noticeable localization errors [1] which motivates the research in HRTF individualization (also referred to as personalization in literature), in order to estimate the individual HRTFs of a subject without having to measure the HRTFs for every individual.

A recent survey [2] summarizes the various methods that have been proposed towards the task of HRTF personalization. Some early works such as [3] aimed at selecting the best fitting HRTF-set from a larger database of HRTFs with anthropometric data as input features. Such approaches would perceptually work only if the user's anthropometric features are close enough to atleast one of the subjects in the database. Hence, some other works later such as [4] used linear regression techniques which provided better results than HRTF selection. Going further, not constraining to linearity, in recent years, several deep learning techniques have been proposed for the individualization of HRTFs. A recent paper [5] provides a great summary of the various deep learning methods that have been devised in this field. Some works such as [6–8] proposed using fully connected networks to predict HRTFs given an input of the anthropometric features, while some other works such as [9, 10] have proposed using perceptual listening feedback to train the network.

### 1.1 Problem Statement A - HRTF Encoding

Using deep learning for predicting HRTFs however, comes with a challenge that the number of available and complete HRTF datasets is limited. To avoid over-fitting then in such situations, it is advantageous to reduce the number of parameters being used in the neural network. This is often achieved by reducing the number of parameters to predict, by encoding the HRTFs into a sub-dimensional space. One approach for dimensionality reduction was proposed by [11], through using Principle Component Analysis (PCA) to project the HRTFs linearly onto dimensions with maximum variance. A suggestion was later reported in [7] that a statistical approach such as PCA fails to encode features that are essential to the individuality of the HRTFs. Some other works such as [12, 13] proposed using fully connected auto-encoders to encode the HRTFs. A fully connected auto-encoder learns the encodings of every single HRTF in space individually, but does not consider the correlations that adjacent HRTFs share in space. Hence, as the first part of our work, we propose encoding the HRTFs by learning the correlations that HRTFs share in space using a Convolutional Neural Networks (CNN) based auto-encoder, hypothesizing that the features encoded in such a latent space would be a function of the body reflections hence making it more specific to every individual.

### 1.2 Problem Statement B - HRTF Interpolation

Most of the previous approaches for HRTF individualization rely on finding a mapping between the anthropometric measurements of individuals to their HRTFs. Having consumers to provide their anthropometric measurements is not necessarily a very practical solution. With the increasing availability of surround sound systems today, a rather futuristic goal would be to measure a sparse set of HRTFs with the available speakers and be able to interpolate to a complete set of HRTFs. Multiple approaches have been proposed previously to solve the problem of HRTF interpolation. Some mathematical approaches such as linear interpolation and thin plate spline interpolation have been previously employed in works such as [14–17]. Some key findings from these works suggest that thin plate spline interpolation of log magnitude HRTFs seems to achieve best interpolation results and that the number of HRTFs in space could be reduced to about $80$ to allow for an accurate reconstruction. Some other works focus on studying the fitting of HRTFs onto the spatially continuous spherical harmonics (SH) domain to allow for a continuous interpolation which was first proposed by [18]. The required order of the SH increases with frequency, and the higher the order required, the more the number of HRTFs are required. It was reported in [19] that for accurate reconstruction of HRTFs upto $20kHz$, about $1600$ HRTFs would be required, less than which leads to spatial aliasing which leads to high-shelf like energy increase in the SH interpolated HRTFs [20]. Some deep learning approaches such as [21, 22] have also been proposed in the past for learning HRTF interpolations, both of which deploy fully connected networks to tackle the problem. Following the success in the image world, where CNN based approaches have provided promising results for interpolation and super resolution as in [23, 24], we propose, as the second part of our work, a CNN based approach for interpolating from a few sparsely distributed HRTFs in space to to the complete HRTF set.

The remainder of the paper is structured as follows. In section 2, we briefly review the theory of HRTFs and our data preparation strategy. In section 3 we describe our proposed model architectures and the training procedures. In section 4, we describe our experiments and discuss the results for the same and finally we provide concluding statements in section 5.

## 2. DATA & DATA PREPARATION

### 2.1 Understanding HRIRs and HRTFs

For a particular point in space, the left and right HRIRs $h_l[n]$ and $h_r[n]$ are defined as the impulse responses for the acoustic system that the sound signal undergoes from its point of origination to the point it reaches the left and right ears respectively. The HRTFs $H_l(\omega)$ and $H_r(\omega)$ are the complex valued frequency domain representations for the left and right HRIRs respectively which can be represented in terms of their magnitude and phase responses as

$$H_{l/r}(\omega) = |H_{l/r}(\omega)|.\angle H_{l/r}(\omega), \tag{1}$$

where the phase response $\angle H_{l/r}(\omega)$ could be represented as,

$$\angle H_{l/r}(\omega) = e^{j\psi(\omega)} \tag{2}$$

From [25], it is possible to derive the minimum phase response $H_{min}(\omega)$ of an HRTF from its magnitude response $|H(\omega)|$. $H(\omega)$ can also be represented in terms of $H_{min}(\omega)$ as

$$H(\omega) = H_{min}(\omega).e^{j\psi_{all}(\omega)}.e^{-j\omega T}, \tag{3}$$

where $e^{\psi_{all}(\omega)}$ is an all-pass phase component and $e^{-j\omega T}$ corresponds to the propagation delay $T$.

Several past works [26–29] have shown that the all-pass component $e^{j\psi_{all}(\omega)}$ could be considered perceptually insignificant upto around $10kHz$. Given then that $H_{min}(\omega)$ could be obtained from $|H(\omega)|$, it is possible to estimate $H(\omega)$ from only its magnitude response and the propagation delay $T$. In this work, we will only focus on estimating $|H(\omega)|$. Estimation of the propagation delays is beyond the scope of this paper.

### 2.2 Database

We used the CIPIC database [30] for this work. The database consists of HRIRs of $45$ subjects, each $200$ samples long, sampled at a sampling rate of $44.1kHz$ and measured at $1250$ points around the head, at a constant radius of $1m$. The positions are sampled in the form of $25$ rings along the azimuth axis and at $50$ points per ring along the elevation axis as shown in Fig. 1.

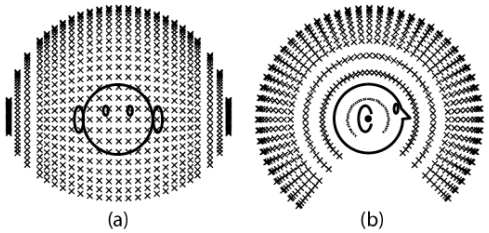



Figure 1. Sampling locations of HRTFs in the CIPIC database (a) front (b) side. The spacing between rings is $5^\circ$ near the center of the head, and increases up to $15^\circ$ towards the side of the head. The $50$ points along every ring are equally spaced, but the spacing is larger towards the center of the head and smaller at the sides of head, to maintain a constant radius.

### 2.3 Data Processing and Arrangement

The HRIRs in the CIPIC database have a dynamic range of $[-2, 2]$. We first obtain the magnitude HRTFs as

$$H_{mag}(\omega) = \left| \frac{2 * FFT\left\{\frac{h[n]}{2}\right\}}{nfft} \right|, \tag{4}$$

where $nfft$, the fft length, is set to $256$. $H_{mag}(\omega)$ is clipped to the minimum at $10^{-6}$ and we pick only the first $128$ bins corresponding to positive frequencies, hence the factor of 2 to account for the energies of the negative frequencies. Equation (4) ensures that $H_{mag}(\omega)$ lies in the range of $[10^{-6}, 1]$. We then obtain the log magnitudes of the obtained magnitude HRTFs and scale them as

$$H_{log}(\omega) = \frac{log_{10}(H_{mag}(\omega))}{6} + 1, \tag{5}$$

hence ensuring that $H_{log}(\omega)$ has a dynamic range of $[0, 1]$.

The $1250$ so obtained HRTF log magnitudes in the database are then arranged in a 3D tensor having a size of $[128$ x $25$ x $50]$ such that the first dimension corresponds to the frequency axis and the second and the third dimensions correspond to the azimuths and the elevations respectively. In this paper to follow, we will refer to the above tensor as the HRTF-tensor.

## 3. PROPOSED MODEL AND TRAINING SPECIFICATIONS

### 3.1 The Model

#### *3.1.1 Part A - AutoEncoder*

The model architecture for our proposed auto-encoder consists of a contracting and expanding path inspired by the very successful U-NET architecture [31]. An example structure having a model depth of 3 is shown in Fig. 2. In this we first pass our input HRTF-tensor through the encoder which consists of multiple convolutional blocks starting with a convolution having a kernel size of $(3, 3)$, stride $(1, 1)$ and a padding of $(1, 1)$ ensuring that the spatial sizes are maintained (will be referred to as the size-maintaining convolutional block from her on). This is followed by several downsampling convolutions having kernel sizes of $(2, 2)$ and strides also of $(2, 2)$, with no $0$ paddings. These kernels hence, both learn the features and also downsample the HRTF-tensor through multiple depth levels along the spatial dimensions. At the final depth level we apply one final size-maintaining convolution to obtain the encoded space. The achieved encoded space is then passed through the decoder which consists of a structure reversing the process carried out by the encoder. In this we start with a size-maintaining convolution followed by several transposed convolutional blocks up the depth levels, with kernel sizes, strides and paddings set appropriately, so as to make sure that the feature maps have the same spatial sizes as the corresponding feature maps of the encoder at the corresponding depth levels. Finally at depth level of 0, we have one final size-maintaining convolutional block that outputs the predicted HRTF-tensor. Just as the convs in the encoder, the transposed convs in the decoder are responsible for both learning the deep features and upsampling in the spatial dimensions. We use the exponential linear units (ELU) activations for all the convolutions and the transposed convolutions, except for the final layer in the decoder predicting the HRTFs, where we use the Sigmoid activation. We also experimented with using batch normalization, and found its effects to be negligible.

Many works have previously adopted such CNN-based auto-encoding structures for various applications, for example [32, 33]. However, our architecture differs from other similar architectures in the following ways.

First, such typical auto-encoding networks employ a number of convolutional layers at every depth level, before using pooling operations for downsampling in the encoder or after the transposed convs in the decoder. Moreover, the number of feature maps or channels double down every depth level. As previously mentioned in section 1, HRTFs come with a challenge of a small dataset, which motivates the use of as few parameters as possible to avoid overfitting. Hence, after multiple experiments, we concluded that the model works best by replacing the pooling operations by the downsampling convolutional blocks, while omitting the other convolutional layers at each depth level, hence allowing for the feature learning to be mostly carried out by the downsampling convs and the transposed convs, while also keeping the number of feature maps constant and equal to the number of frequency bins, all throughout.

Secondly, typically in 2D CNNs, each filter is expected to learn the same spatial features for all channels. We hypothesized, as also suggested in [9], that the spatial correlations of the HRTFs are expected to be different along the frequency axis. Hence we used the "groups" option in PyTorch's Conv2D function and we set it to $8$. This splits the tensors along the channels axis into 8 groups before the convolution operation. Each filter now only learns the correlations of a single group, hence allowing for the learning of different features for different frequency bins. We found that this approach significantly improved the accuracy of

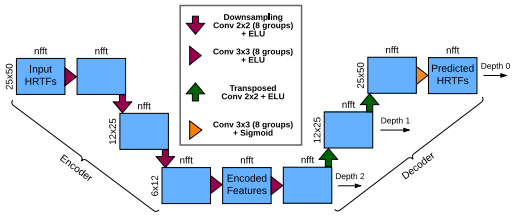



Figure 2. HRTF auto-encoder model architecture. The number above the blue blocks corresponds to the number of channels/feature maps. The number to the left of the blue blocks corresponds to the spatial size of the blocks at every depth level. The spatial sizes remain constant along every depth level. Note that nfft here corresponds to the number of positive frequency bins.

our model.

Additionally, one of the key components of a U-NET structure is the residual connections between the feature maps of the encoder to the feature maps of the corresponding depth level in the decoder. Although we concur that using residual connections does improve the accuracy of the auto-encoding structure itself, we reckon that applications of such an auto-encoding process lie in the possibility of predicting the encoded space through other means, and then using the decoder to obtain the HRTFs, an example of which we show in the work to follow. In such a case, we would not have access to the encoder during inference, and therefore we propose not using residual connections between the encoder and the decoder in this work.

#### *3.1.2 Part B - Interpolation*

For the next part of our work, we propose a transfer learning like approach for HRTF interpolation. In this, we use the learned decoder from the auto-encoder network and train a new encoder (will be referred to as the sparse-encoder from here on) to map the sparse HRTF-tensor onto the encoded space obtained using the auto-encoder. The flow diagram for our model is shown in Fig. 3. The encoding network for the sparse-encoder comprises of a single convolutional layer, parameters of which are decided by the spatial size of the input sparse HRTF-tensor that we wish to interpolate. In this work, we demonstrate the interpolation for two cases – 1. that of a sparse HRTF-tensor having a spatial size of 6 x 12, and 2. that of a sparse HRTF-tensor having size 3 x 6.

For case 1 we use for the encoder, a single convolutional layer with kernel size $(3, 3)$, stride $(1, 1)$ and padding of $(1, 1)$. For case 2 we use a single transposed convolution layer with kernel size $(2, 2)$ and stride of $(2, 2)$ with no $0$ padding. We use the ELU activation for both cases since the decoder was trained to decode from an encoded space which had gone through an ELU activation itself.

The decoder structure for both cases is the same as that of the auto-encoder with a model depth of 3.

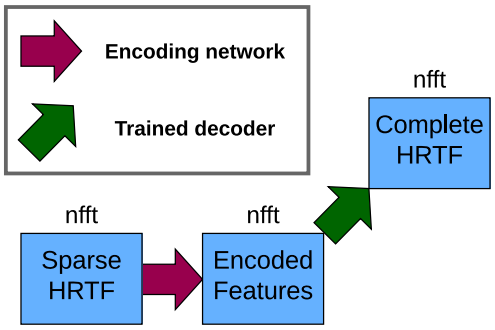



Figure 3. Flow diagram for interpolation of sparse HRTF

### 3.2 Training Specifications

Of the $45$ subjects available in the CIPIC database, we randomly set 3 subjects aside as test subjects and trained our models on the remaining $42$ subjects. For both the auto-encoding and the interpolation networks, we use full-batch training using the ADAM optimizer with an initial learning rate of $0.0001$. We use the early stopping strategy where we let our training process run for a maximum of $400000$ epochs and stop the training when the validation error starts to go up while picking the model with the lowest validation loss. More sophisticated validation strategies such as leave-one-out cross validation, which is often suited for small datasets, but which requires significant resources in terms of the total training time, remain to be a future work at the moment.

For the interpolation problem, we first trained the sparse-encoder while freezing the weights of the pre-trained decoder from the autoencoder model, until the validation error reached a plateau. We then unfroze the weights of the decoder while setting the learning rate of the decoder to $\frac{1}{100}^{th}$ that of the sparse-encoder and continued the training process following the early stopping procedure. We found that this scheme of fine-tuning the weights of the pre-trained decoder to provide better results as opposed to training the decoder and the sparse encoder end-to-end, or as opposed to not fine-tuning the pre-trained decoder at all.

After experimentation, we found that use of dropouts and regularization schemes to not be very effective for these experiments and they only prolonged the training process, without actually resulting in better reconstructions.

### 3.3 Loss Function

We used a mean Log Spectral Distortion (LSD) function as both the loss function to train our models and the evaluation criteria. For a pair of ground truth and predicted log magnitude HRTFs $H$ and $\hat{H}$ respectively, the LSD can be defined as

$$LSD(H, \hat{H}) = \sqrt{\frac{1}{k_2 - k_1 + 1} \sum_{k=k_1}^{k_2} \left(\hat{H}(k) - H(k)\right)^2}, \tag{6}$$

where the frequency bin numbers $k_1$ and $k_2$ correspond to the starting and ending frequency bins respectively along which to compute the LSD. We used full band LSD to evaluate our models, in other words setting $k_1$ to 0 and $k_2$ to 127.

## 4. EXPERIMENTS, RESULTS & DISCUSSION

Before moving on to the experiments, we would like to give a few details about the LSD plots that we use to report our results.

### 4.1 Understanding the LSD Plots

The LSD plot demonstrates the LSD values of the reconstructed HRTFs for a particular subject at all points in space available in the database, in our case 1250 points. We use the inter-aural polar co-ordinate system to label the azimuth $\theta$ and the elevation $\phi$ angles. In this, the points to the left of the head are found at azimuths $80° > \theta > 0°$ whereas points to the right of the head are found at $0° > \theta > -80°$ with $\theta = 0°$ being exactly in front or back of the head. Points in front of the head are found at elevations $-45° < \phi < 90°$ while points at the back of the head are found at $90° < \phi < 230°$ with $\phi = 0°$ and $\phi = 180°$ corresponding to the lateral plane at ear level and $\phi = 90°$ being the point exactly above the head. Thereby, $-45° < \phi < 0°$ and $230° > \phi > 180°$ correspond to points below the head and $0° < \phi < 180°$ correspond to points above the head.

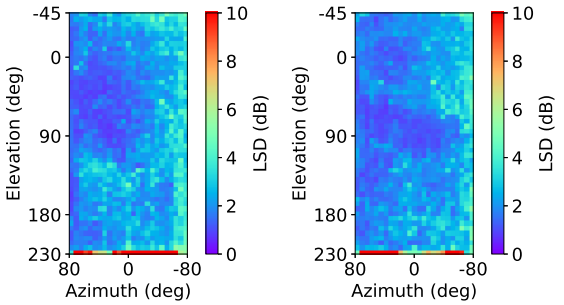
In the remainder of this section, we report the results of our auto-encoder and interpolation models. For brevity, we will only be reporting results for the predictions of the log magnitude HRTFs of the left ear.

### 4.2 Experiment 1 - AutoEncoding

The aim of this experiment was to study the reconstruction capability of our auto-encoder model described in section 3.1.1 for model depths of 3 and 4. A model depth of 3 encodes the HRTF-tensor, which consists of 25x50 HRTFs, to an encoded space having a spatial size of 6x12 and as many channels as the number of frequency bins, in our case 128. A model depth of 4 encodes the HRTF-tensor one step further to an encoded space of size 128x3x6.

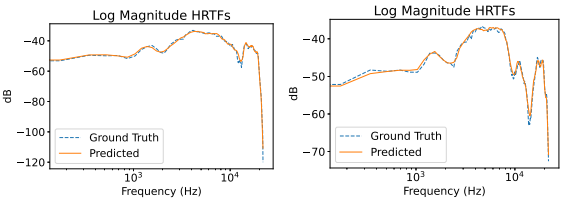
Fig. 4 shows the LSD plot for 2 test subjects for a model depth of 3. Fig. 5 compares the reconstructed HRTFs with the ground truth HRTFs for the point located at [$\theta = 45°$, $\phi = 45°$], or in other words the point at $45°$ to the left and above in front of the head.

The LSD plots suggest that the reconstruction is better in the area above the head (the upper half of the lateral plane) and on the ipsilateral side of the HRTFs (in this case the left side for left HRTFs). There is a significant drop in the performance for points on the extreme bottom behind the head and also at some points on the extreme right side. We suspect this error to be related to either the 0 padding schemes in our convolutional layers, or bad points in the dataset itself, and further investigation on this anomaly is left for future work.




(a) Subject 0 (b) Subject 1

Figure 4. LSD plots for the AutoEncoder predictions for a model depth of 3 for 2 test subjects (left ear)

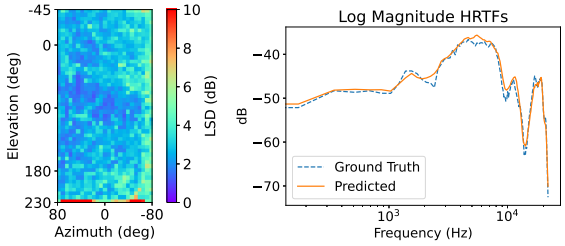



(a) Subject 0 (b) Subject 1

Figure 5. HRTF plots for the AutoEncoder predictions for a model depth of 3 for 2 test subjects at an azimuth and elevation of $45°$ on the ipsilateral side of the left ear

The HRTF plots reveal that although the HRTFs for the two subjects at the same location differ significantly from one another, the model was able to reconstruct these to a great degree, hence indicating that it successfully encodes individuality in the HRTFs.

Fig. 6 shows the LSD plot and the HRTF plot for 1 subject for a model depth of 4. As expected, the reconstruction errors are higher in this case as a result of encoding the HRTF-tensor to an even smaller space. The HRTF plot suggests that this results in a rather more smoothed HRTF reconstruction. Although, whether this difference in the reconstruction is audible could only be answered through perceptual listening tests, which is left for future work. Table 1 compares the mean LSD values and their standard deviations across all 3 test subjects and for all points in space for the two cases. Interestingly, the standard deviations for both models are exactly the same. This suggests that the correlations between certain points in space are being better learned than at other points, irrespective of the model depth. Given that CNNs assume an equivalent correlation across all points in space, splitting the HRTF-tensors into multiple regions spatially and applying a different network to each region might provide better results and is left for future work. Furthermore, it might also be worthwhile to obtain results on some other HRTF databases where the points are sampled at spherically equidistant locations such as the ARI database [34].




(a) LSD plot (b) HRTF plot

Figure 6. LSD and HRTF plots for AutoEncoder with a model depth of 4 (left ear)

| Depth Level | Mean $\pm$ Std LSD (dB) |
|---|---|
| 3 | 2.38 $\pm$ 1.52 |
| 4 | 3.24 $\pm$ 1.52 |

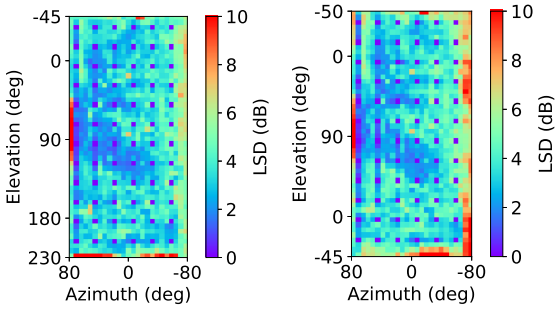
Table 1. Comparing mean and std LSD values for AutoEncoder with model depths 3 v/s 4. Note that the standard deviations are calculated across HRTFs at all positions in space and for all the test subjects

### 4.3 Experiment 2 - Interpolation

To obtain the sparse HRTF-tensor, we sampled the complete HRTF-tensor uniformly. We obtained 2 such sparse HRTF-tensors. For the first one, we sampled at every 4 points in both the azimuth and elevation dimensions, starting from index 1 for azimuth ($[1, 5, 9...]$) and starting from index 2 for the elevations ($[2, 6, 10, ...]$), thereby obtaining a sparse HRTF-tensor having a size of 6x12. For the second one, we sampled at every $8^{th}$ point starting at index 4 in both dimensions ($[4, 12, ...]$), thereby obtaining a sparse HRTF-tensor with size 3x6. We will refer to these sparse HRTF-tensors as sparse-tensor-1 and sparse-tensor-2 respectively for the discussion to follow. After obtaining our predicted complete HRTF-tensors, we re-inserted the corresponding sparse HRTF-tensors in the predictions at the appropriate positions.

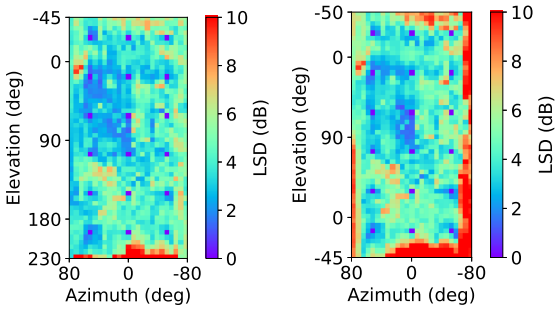
To report our results, we compare our proposed interpolation strategy with that obtained using bilinear interpolation. Bilinear interpolation only allows for interpolation between points inside the boundaries of the sparse-tensor and is incapable of extrapolating points outside the boundaries. However, the CNN model provides both interpolation as well as extrapolation. For visualization purposes, in the bilinear interpolation scheme, we copied the HRTFs at the boundary to substitute for the unavailable points outside the boundaries.

Figs. 7 and 8 show the LSD plots for the interpolations of sparse-tensor-1 and sparse-tensor-2 respectively. We can see some improvement in our proposed model with respect to the bilinear interpolation scheme, especially in the region above the head and on the ipsilateral side. The improvement is more apparent for sparse-tensor-2. We can also see that the method does a fair job in extrapolating points outside the selected region in the front of the head and on the ipsilateral side. We achieved a mean LSD




(a) Proposed Model (b) Bilinear interpolation

Figure 7. LSD plots for the interpolation of sparse-tensor-1 (left ear)




(a) Proposed Model (b) Bilinear interpolation

Figure 8. LSD plots for the interpolation of sparse-tensor-2 (left ear)

of $3.34dB$ and $4.12dB$ for the interpolation and extrapolation of sparse-tensor-1 and sparse-tensor-2 respectively. Table 2 provides the LSD comparisons for all test subjects over all the locations between the proposed model and the bilinear interpolation scheme. To allow for a fair comparison, we only report the LSDs for the interpolation of points inside the boundary in this table. The LSD results show that the proposed model provides about 10% improvement over the bilinear interpolation scheme.

| Model | Sparse-tensor-1 Mean $\pm$ Std LSD | Sparse-tensor-2 Mean $\pm$ Std LSD |
|---|---|---|
| CNN | 2.82 $\pm$ 1.29 | 3.61 $\pm$ 1.32 |
| Bilinear | 2.99 $\pm$ 1.37 | 3.95 $\pm$ 1.39 |

Table 2. Comparing the mean and std LSD values for interpolation by the proposed CNN model v/s bilinear interpolation

## 5. CONCLUSION

In this work, we presented an alternate strategy for the encoding of HRTFs using a CNN-based auto-encoder. We showed that an HRTF-set consisting of 1250 locations could be encoded to a latent space having a spatial size

of 6x12 with a mean LSD of $2.38dB$ and to a space having a size of 3x6 with a mean LSD of $3.24dB$. We further showed that such an encoded space could be used towards learning interpolations from an HRTF-set sparsely sampled in space. In doing so, we showed that a sparse HRTF-set sampled at 72 locations results in a mean reconstruction error of $3.34dB$ LSD and that sampled at 18 locations results in a mean reconstruction error of $4.12dB$ LSD, both cases also supported by the LSD plots providing more insight into the LSD errors per location. The results also suggested that such an interpolation scheme allows for not only interpolation, but also extrapolation to some extent.

Some immediate next steps in this work remain to be trying this algorithm on other available datasets to obtain a more generalized result, while also using more sophisticated cross-validation strategies such as the leave-one-out strategy. Finally, it would be crucial to conduct perceptual listening tests to understand the perceptual effects of the proposed approach.

Having said that, the above presented results provide plenty insights into suggesting that it is indeed possible to encode HRTFs using convolutional neural networks using only 42 datapoints while allowing the encoding of features essential for the individuality of the HRTFs. This leads way to finding mappings between other features that define the HRTFs and the learned encoded space, for example, using using sparsely sampled HRTFs in space as shown in this work. We showed that the encoded space could be leveraged to interpolate HRTFs to an acceptable degree with only 18 locations. With the extent of surround sound systems that are available today such as the 11.1.8 system which consists of 19 speakers, we believe as a rather futuristic goal, that this work might be the first step into finding mappings between the latent representations of the HRTFs measured using these speakers (although corrupted with room reflections) to the latent space of the user's clean and complete HRTFs, allowing for the prediction of the complete clean HRTF set using the corrupted sparse HRTF measurements obtained using the surround sound speakers for example. We hope that such mappings between latent features of similar acoustic systems might open up interesting avenues.

## 6. REFERENCES


[1] E. M. Wenzel, M. Arruda, D. J. Kistler, and F. L. Wightman, "Localization using nonindividualized head-related transfer functions," *The Journal of the Acoustical Society of America*, vol. 94, no. 1, pp. 111–123, 1993.

[2] C. Guezenoc and R. Seguier, "Hrtf individualization: A survey," *arXiv preprint arXiv:2003.06183*, 2020.

[3] D. Zotkin, J. Hwang, R. Duraiswaini, and L. S. Davis, "Hrtf personalization using anthropometric measurements," in *2003 IEEE workshop on applications of signal processing to audio and acoustics (IEEE Cat. No. 03TH8684)*. Ieee, 2003, pp. 157–160.

[4] P. Bilinski, J. Ahrens, M. R. Thomas, I. J. Tashev, and J. C. Platt, "Hrtf magnitude synthesis via sparse representation of anthropometric features," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2014, pp. 4468–4472.

[5] R. Miccini and S. Spagnol, "Hrtf individualization using deep learning," in *2020 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*. IEEE, 2020, pp. 390–395.

[6] C. J. Chun, J. M. Moon, G. W. Lee, N. K. Kim, and H. K. Kim, "Deep neural network based hrtf personalization using anthropometric measurements," in *Audio Engineering Society Convention 143*. Audio Engineering Society, 2017.

[7] H. Fayek, L. van der Maaten, G. Romigh, and R. Mehra, "On data-driven approaches to head-related-transfer function personalization," in *Audio Engineering Society Convention 143*. Audio Engineering Society, 2017.

[8] G. W. Lee and H. K. Kim, "Personalized hrtf modeling based on deep neural network using anthropometric measurements and images of the ear," *Applied Sciences*, vol. 8, no. 11, p. 2180, 2018.

[9] K. Yamamoto and T. Igarashi, "Fully perceptual-based 3d spatial sound individualization with an adaptive variational autoencoder," *ACM Transactions on Graphics (TOG)*, vol. 36, no. 6, pp. 1–13, 2017.

[10] Y. Shu-Nung, T. Collins, and C. Liang, "Head-related transfer function selection using neural networks," *Archives of Acoustics*, vol. 42, no. 3, pp. 365–373, 2017.

[11] H. Hu, L. Zhou, H. Ma, and Z. Wu, "Hrtf personalization based on artificial neural network in individual virtual auditory space," *Applied Acoustics*, vol. 69, no. 2, pp. 163–172, 2008.

[12] T.-Y. Chen, T.-H. Kuo, and T.-S. Chi, "Autoencoding hrtfs for dnn based hrtf personalization using anthropometric features," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 271–275.

[13] Y. Luo, D. N. Zotkin, and R. Duraiswami, "Virtual autoencoder based recommendation system for individualizing head-related transfer functions," in *2013 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*. IEEE, 2013, pp. 1–4.

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

[15] S. Carlile, C. Jin, and V. Van Raad, "Continuous virtual auditory space using hrtf interpolation: Acoustic and psychophysical errors," in *Proceedings of the First IEEE Pacific-Rim Conference on Multimedia*, 2000, pp. 220–223.

[16] P. Minnaar, J. Plogsties, and F. Christensen, "Directional resolution of head-related transfer functions required in binaural synthesis," *Journal of the Audio Engineering Society*, vol. 53, no. 10, pp. 919–929, 2005.

[17] R. Martin and K. McAnally, "Interpolation of head-related transfer functions," DEFENCE SCIENCE AND TECHNOLOGY ORGANISATION EDINBURGH (AUSTRALIA) AIR . . . , Tech. Rep., 2007.

[18] M. J. Evans, J. A. 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, pp. 2400–2411, 1998.

[19] W. Zhang, T. D. Abhayapala, R. A. Kennedy, and R. Duraiswami, "Insights into head-related transfer function: Spatial dimensionality and continuous representation," *The Journal of the Acoustical Society of America*, vol. 127, no. 4, pp. 2347–2357, 2010.

[20] J. M. Arend, F. Brinkmann, and C. Pörschmann, "Assessing spherical harmonics interpolation of time-aligned head-related transfer functions," *Journal of the Audio Engineering Society*, vol. 69, no. 1/2, pp. 104–117, 2021.

[21] V. Lemaire, F. Clerot, S. Busson, R. Nicol, and V. Choqueuse, "Individualized hrtfs from few measurements: a statistical learning approach," in *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, vol. 4. IEEE, 2005, pp. 2041–2046.

[22] G. Kestler, S. Yadegari, and D. Nahamoo, "Head related impulse response interpolation and extrapolation using deep belief networks," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 266–270.

[23] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1125–1134.

[24] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang *et al.*, "Photo-realistic single image super-resolution using a generative adversarial network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4681–4690.

[25] A. Oppenheim, R. Schafer, and J. Buck, "Discrete-time signal processing: Prentice-hall englewood cliffs," 1989.

[26] S. Mehrgardt and V. Mellert, "Transformation characteristics of the external human ear," *The Journal of the Acoustical Society of America*, vol. 61, no. 6, pp. 1567–1576, 1977.

[27] A. Kulkarni, S. Isabelle, and H. Colburn, "On the minimum-phase approximation of head-related transfer functions," in *Proceedings of 1995 Workshop on Applications of Signal Processing to Audio and Acoustics*. IEEE, 1995, pp. 84–87.

[28] D. J. Kistler and F. L. Wightman, "A model of head-related transfer functions based on principal components analysis and minimum-phase reconstruction," *The Journal of the Acoustical Society of America*, vol. 91, no. 3, pp. 1637–1647, 1992.

[29] A. Kulkarni, S. Isabelle, and H. Colburn, "Sensitivity of human subjects to head-related transfer-function phase spectra," *The Journal of the Acoustical Society of America*, vol. 105, no. 5, pp. 2821–2840, 1999.

[30] V. R. Algazi, R. O. Duda, D. M. Thompson, and C. Avendano, "The cipic hrtf database," in *Proceedings of the 2001 IEEE Workshop on the Applications of Signal Processing to Audio and Acoustics (Cat. No. 01TH8575)*. IEEE, 2001, pp. 99–102.

[31] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.

[32] Ö. Çiçek, A. Abdulkadir, S. S. Lienkamp, T. Brox, and O. Ronneberger, "3d u-net: learning dense volumetric segmentation from sparse annotation," in *International conference on medical image computing and computer-assisted intervention*. Springer, 2016, pp. 424–432.

[33] P. Chandna, M. Blaauw, J. Bonada, and E. Gómez, "Wgansing: A multi-voice singing voice synthesizer based on the wasserstein-gan," in *2019 27th European Signal Processing Conference (EUSIPCO)*. IEEE, 2019, pp. 1–5.

[34] Ari database webpage. [Online]. Available: https://www.oeaw.ac.at/en/isf/das-institut/software/hrtf-database