



Master MVA: 3D Sound Rendering (Lab)

Gaël RICHARD

26 février 2026

In this Lab, we propose to program a spatializer for binaural listening (e.g. using headphones). To that aim, we will use a dataset of HRTFs or more precisely HRIRs (Head Related Impulse responses). Each HRIR is sampled at 44.1 kHz and depends on azimuth, elevation and time. The dataset was recorded by the CIPIC Interface Laboratory of University of California [1].

1 The database of HRTFs (or HRIRs)

In the datafile, HRIRs are stored in two arrays containing all HRIRs recorded at discrete positions in space. Then, the azimuth, elevation and time values are respectively specified by the discrete indices naz , nel and nt . (naz, nel, nt) is then an array of $25 \times 50 \times 200$ (25 azimuths, 50 elevations and 200 time samples).

The azimuth is an angle θ defined between the mid-sagittal plane and the vertical plane containing the head center and the considered position. Values are then between -90° and $+90^\circ$ (Values are not equally spread since there are a greater density of points between -45° and $+45^\circ$).

The elevation is an angle defined between the horizontal plane and the plane containing the source and the head center. Values are given by steps of 5.625 between -45 and $+230$, knowing that there are no values between -90 and -45 and between 230 and 270 .

With these coordinates, we have the following correspondences:

- $(0,0)$ is a point which is right in front (at 1 m)
- $(0,90)$ is a point which is right above (at 1 m)
- $(0,180)$ is a point which is right behind (at 1 m)
- $(90,0)$ is a point which is straight on the right (at 1 m)
- $(-90,0)$ is a point which is straight on the left (at 1 m)

2 Database visualization

In this first part, the goal is to better understand the content of the HRTF database.

To that aim, it is proposed to use the provided python script *TP-spatialisation-showHRTFdata.ipynb* to visualize the HRTF database.

This script allows you to visualize, for a given subject :

- all HRTF stacked in a matrix for the left and right ears with time in y-coordinate and angle in x-coordinate for a fixed elevation angle of 0 degree ;
- the corresponding frequency responses of all HRTF stacked in a matrix with frequency in y-coordinate and angle in x-coordinate;
- all HRTF stacked in a matrix for the left and right ears with time in y-coordinate and elevation angle in x-coordinate for a fixed azimuth angle of 0 degree ;
- the corresponding frequency responses of all HRTF stacked in a matrix with frequency in y-coordinate and angle in x-coordinate;
- The impulse response of one HRTF for a given position and subject for both ears.

Questions

1. What is the role of the function *getNearestUCDpulse*?
2. Run the script of a given subject (for instance subject 3). Explain why the HRTFs are varying with the angle and with elevation.
3. Run the script for different subjects (*4 subjects are available: subject03, subject044, subject065 and subject165*) . Do you see differences between subjects? Can they be explained?
4. Why the displayed HRIRs at the end of the script have the same delay?

3 3D sound rendering with HRTFs

In this part, we propose to program a spatializer for binaural listening (e.g. using headphones). To that aim, we will use the CIPIC HRTF dataset.

3.1 Problem 1: Binaural filtering

In this problem, we will program a function called *hrtf_binaural* which will simulate a binaural system (see figure 1). This program will be tested on several sound trajectories to be defined.

3.1.1 Binaural filtering for a fixed position in space

- Choose a subject number and load its corresponding HRIR datasets which are stored in a file *.mat*
- Use the provided program *GetNearestUCDpulse(azimuth,élévation,h3D)*, to obtain the HRIRs corresponding to the predefined fixed source position in space.
- load one of the sound example provided in the sound directory.
- Compute the binaural signals y_g and y_d by filtering the original signal with the corresponding HRIR for left and right ears and build the stereo signal. Listen and comment the results.

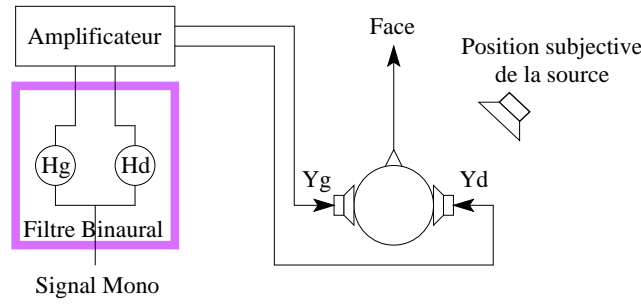


FIG. 1 – *binaural signal synthesis*

3.1.2 Binaural filtering for a trajectory

- Define a sound trajectory in space (e.g. on the sphere of radius 1 meter around the head)
- Synthesize and listen the sequence of sounds placed along this trajectory. For the sake of simplicity, choose a brief sound (for example 'snare1_44.wav') and play the sound at a single position until its end, and then re-play the sound at its new location, and so on...
- What could you do to have a sound that moves continuously in space?
- Compare this approach with the simple "panpot"¹? What are the main differences? do you have a preference? are the provided HRTF optimal? Justify your answer and propose solutions to improve this system in general, and in particular for you.

3.2 Problem 2: Transaural filtering

Transaural filtering consists in transposing the previous system for a loudspeaker setting. To that aim, it is necessary to transform the binaural signal to take into account the contribution of each loudspeaker on each ear.

The figure 2 gives a simple implementation of binaural signal transcoding into a transaural signal.

The goal of this problem is then to program such a system.

- First, express the filters $H'_{gg}, H'_{dd}, H'_{dg}, H'_{gd}$ from the filters $H_{gg}, H_{dd}, H_{dg}, H_{gd}$ corresponding to the transfer functions between a given loudspeaker and one ear (for instance H_{gd} corresponds to the transfer function between the left loudspeaker and the right ear)
- To realize the transaural system, one may inverse the filter $H_{den} = H_{gg}H_{dd} - H_{dg}H_{gd}$ using the least square technique. Let $G(z) = H(z)^{-1}$, we have $d(n) = h_{den}(n) * g(n) = \sum_{k=0}^{N-1} h_{den}(k)g(n-k)$. Then, writing this equation for n in the range $[-A, C+2]$, one

1. we recall that the stereo panpot method consists in computing a stereo signal using the following formula and where $x(n)$ is the original monophonic signal:

$$D = (1 - \alpha)x(n) \quad (1)$$

$$G = \alpha x(n) \quad (2)$$

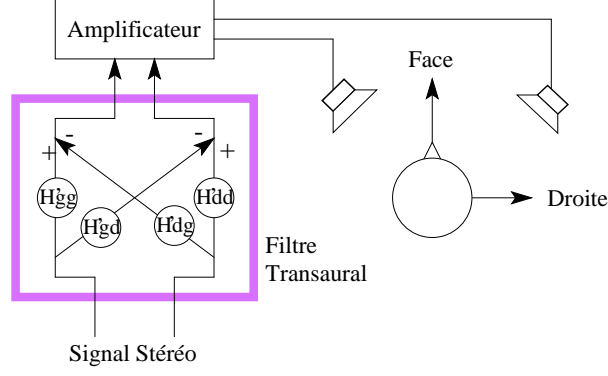


FIG. 2 – *Transaural rendering for a loudspeaker setting*

obtains (for instance for the case $N = 3$):

$$\left\{ \begin{array}{lcl} d(-A) & = & h(0)g(-A) \\ d(-A+1) & = & h(1)g(-A) + h(0)g(-A+1) \\ d(-A+2) & = & h(2)g(-A) + h(1)g(-A+1) + h(0)g(-A+2) \\ \vdots & = & \vdots \\ d(0) & = & h(2)g(-2) + h(1)g(-1) + h(0)g(0) \\ \vdots & = & \vdots \\ d(C) & = & h(2)g(C-2) + h(1)g(C-1) + h(0)g(C) \\ d(C+1) & = & h(2)g(C-1) + h(1)g(C) \\ d(C+2) & = & h(2)g(C) \end{array} \right. \quad (3)$$

To guarantee that $g(n)$ represents the impulse response of the inverse filter, it is necessary that $d(n)$ verifies $d(n) = (0, 0, \dots, 0, 1, 0, \dots, 0)$. The previous equations can be rewritten in matricial form as $\mathbf{d} = \mathbf{H}\mathbf{g}$ where \mathbf{H} is a Toeplitz matrix built from $h(0), h(1), \dots, h(N-1)$. A solution is then given by the pseudo inverse $\mathbf{g} = \mathbf{H}^\dagger \mathbf{d}$. (for potential programming in Matlab, one can use the function `toeplitz` and operator `\`).

Références

- [1] V. R. Algazi, R. O. Duda, D. M. Thompson, and C. Avendano. The cipic hrtf database. In *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pages 99–102, 2001.