Spatializer for Virtual Loudspeakers

Objective

Implement a system that can position a mono sound source in auditory space on the horizontal plane using Vector Based Amplitude Panning (VBAP). It is achieved through the use of Head Related Impulse Responses (HRIRs) to establish virtual speakers and then positioning the virtual source by modifying the gain factors of the chosen speaker pair.

Motivation

Intensity panning is used in mixing consoles and Digital Audio Workstations (DAWs) to pan between speakers. VBAP extends this concept to position an audio source in a virtual speaker setup by calculating the gain values of the speaker pair between which it is to be placed. Such a spatializer system could be used to create rudimentary spatial compositions for virtual speaker playback over headphones. It could also easily be adapted to create for a multi-channel loudspeaker setup as well.

Concept

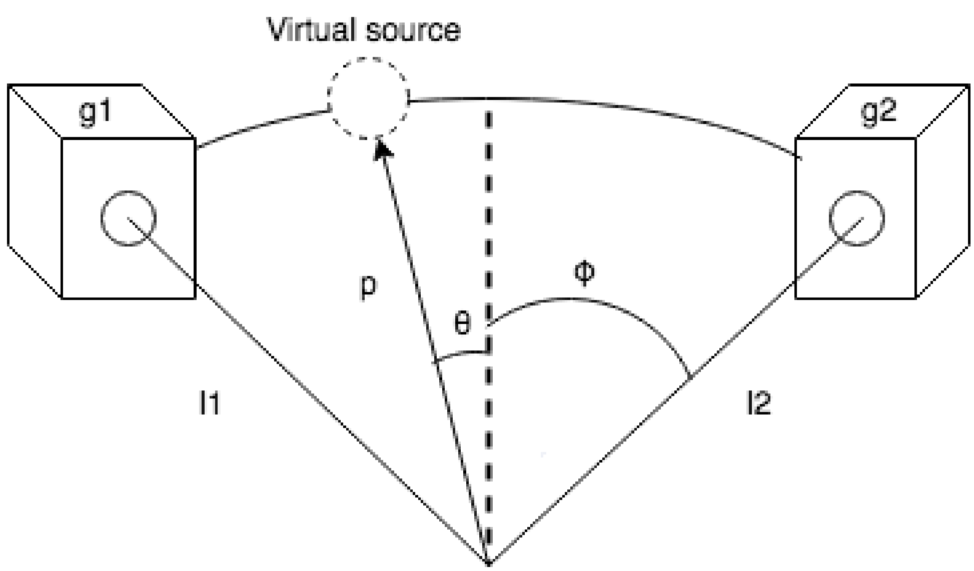

Figure 1. VBAP

Given the position vectors of two virtual speakers and the position vector of the virtual source, the required gain values of the speakers can be calculated as follows:

$P = G \cdot L_{12}^{-1}$

Where P- position vector of virtual source,

G- gain vector of speaker pair

L_{12}- position matrix of speaker pair as seen in Figure 1.

Program flow

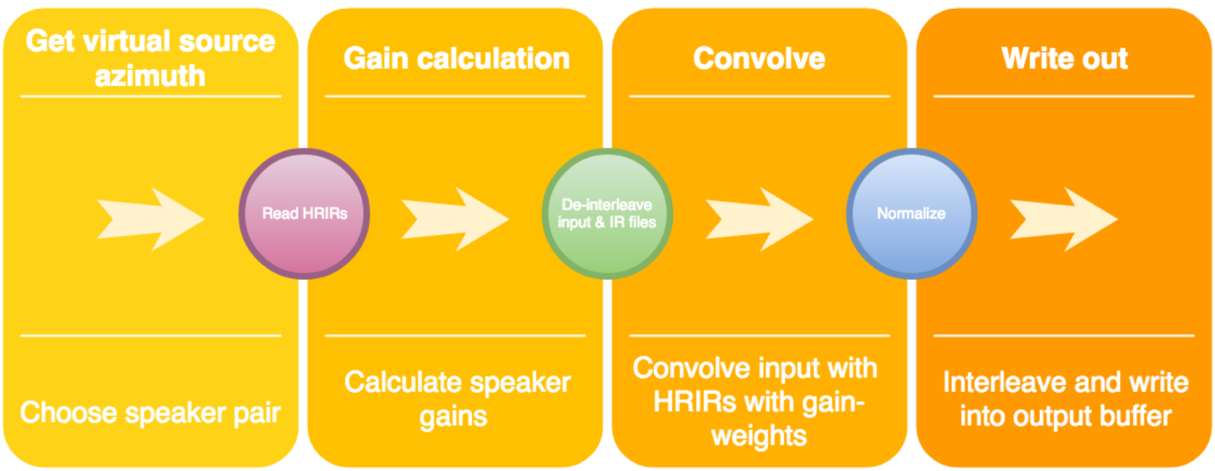

Figure 2. Program flowchart

In this program, all the code was written by me with help from Libsndfile library functions to read and write sound files.

The main functional segments are as follows (Figure 2):

1. Get azimuth user input
2. Choose appropriate speaker pair
3. Read appropriate HRIR wav file(s)- uses Libsndfile
4. Calculate speaker gains for desired azimuth- in findGain() function
5. De-interleave input and HRIR files for convolution

6. Convolve gain-weighted input with HRIR
7. Normalize for equal loudness across different azimuths
8. Interleave and write into output buffer- uses Libsndfile

Most of the processing is in the main() function.

Complexity

The main processing is the convolution, which is done in the time-domain. Thus, the overall complexity of the program is $O(n^2)$.

Platform

The program was implemented on Mac OSX, using the Terminal.

The following command line statements can be used to build the program file-

```
gcc finalProject.c -o finalProject -I/usr/local/include \-
L/usr/local/lib –lsndfile
```

Since Libsndfile is used in the program, it is included in the build statement.

The program does not take any command line arguments, and takes one user input per execution since it is not in real time. It can be executed with the following command line statement-

```
./finalProject
```

Assumptions

The program assumes that the input file is the *mono* file *'svega_44.wav'*.

The program also assumes that the HRIR wav files are in the current directory of execution. This is seen in the path name for the HRIR files.

Future work

1. The system needs to be implemented in real-time so the user can position the sound at different positions without having to re-execute the program.

2. Implement 3D VBAP that can spatialize the target sound in all directions around the listener.
3. The presence of perceptible level differences between positions on the left and right sides of the median plane needs to be investigated further. Some users of the program reported a lower level on the right-side positions (such as 90º) compared to right side positions such as 270º. This could be due to the differences between the left and right ears of the dummy head used in the measurements, but this has not been verified.

References

Pulkki, V. (1997). Virtual sound source positioning using vector base amplitude panning. *Journal of the Audio Engineering Society*, *45*(6), 456-466.