

SOFA Format für HRIRs und BRIRs

Um verschiedene HRIRs und BRIRs benutzen zu können und diese auch einfach mit anderen teilen zu können, wurde SOFA (<https://www.sofaconventions.org>) als gemeinsames Dateiformat eingeführt. Für Matlab bringt es bereits eine API mit, die einen einfachen Zugriff auf die Daten erlaubt, siehe https://github.com/sofacooustics/API_M0. Ein paar Beispiele für eine Verwendung der API sind unter <http://docs.twoears.eu/en/latest/database/impulse-responses/usage/> zu finden (jeweils ab dem `SOFAload`-Befehl). Weitere nützliche Funktionen zum Umgang mit Impulsantworten befinden sich zudem in der Sound Field Synthesis (SFS) Toolbox (<https://github.com/sfstoolbox/sfs-matlab>).

- 1) Lade die SOFA API für Matlab herunter und installiere sie. Installiere zudem die momentane Master-Version der SFS Toolbox.
- 2) Verschaffe Dir einen Überblick über die in den beiden Dateien `HRIRs.sofa` und `BRIRs.sofa` gespeicherten Impulsantworten (wie viele Lautsprecher, welche Lautsprecherpositionen, welche Zuhörerposition, welche Kopfdrehungen, ...)
- 3) Falte jeweils das Signal `flamenco.wav` mit einem HRIR-Paar und einem BRIR-Paar, so dass es jeweils von vorne wahrgenommen wird. Höre Dir beide Ergebnisse an und vergleiche sie.

Bei der Binauralsynthese kann es zu Beeinträchtigungen des Frequenzgangs kommen. Dieses ist zum einen durch die verwendeten Kopfhörer als auch durch die Position der Mikrofone bei der Aufnahme bedingt. Frequenzgänge von Kopfhörern können tiefe Dips aufweisen, die sich zudem bei Auf- und Absetzen der Kopfhörer leicht verschieben können. Es ist daher nicht möglich diese komplett auszugleichen. In der Praxis begnügt man sich meist mit generischen Filtern, die in den meisten Fällen zu guten Ergebnissen führen.

Für die folgenden Aufgaben sind die Funktionen `spectrum_from_signal` und `convolution` aus der SFS Toolbox von Interesse.

- 4) Lade den Entzerrungsfilter `eq_filter.wav` und vergleiche dessen Frequenzgang mit dem Frequenzgang der HRIR von vorne.
- 5) Wende den Filter auf die oben erzeugten Signale an und vergleiche das Ergebnis.

Vergleich Panning und Binauralsynthese

Ziel der Binauralsynthese ist es Hörereignisse außerhalb des Kopfes zu ermöglichen. Dieses ist mit den üblichen Panningverfahren meistens nicht möglich. Im Folgenden wollen wir beide Verfahren miteinander vergleichen.

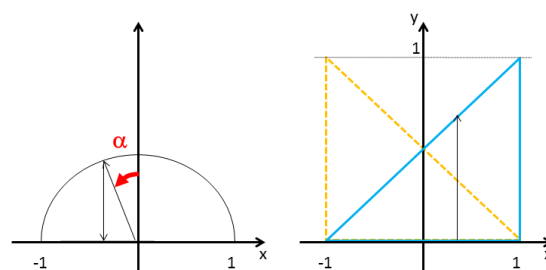


Abbildung 1: Geometrie zur Pegeldifferenz-Stereophonie. Links: Abbildung von Winkel α auf x-Achse. Rechts: Lineare Verstärkungsfunktionen für linken und rechten Kanal in Abhängigkeit von x . Es gilt ferner: $x < 0 \hat{=}$ links, $\alpha > 0 \hat{=}$ links.

- 6) Erstelle eine Funktion `intensity_stereo.m`, die es ermöglicht, ein Monosignal mittels der Pegeldifferenz-Stereophonie für verschiedene Positionen, sprich für verschiedene Winkel α als Stereosignal wiederzugeben. Verwende dazu die geometrischen Vorgaben aus Abb.1.

Hinweis: Rechne zunächst den Winkel α auf eine Position in x -Richtung (Abb.1 links) und berechne Sie dann für diese Position x die Verstärkungsfaktoren y_{links} und y_{rechts} für den jeweils linken und rechten Kanal (Abb.1 rechts).

- 7) Die Funktion `time_stereo.m` enthält eine Implementation der Laufzeit-Stereophonie. Vergleiche deren Ergebnisse mit der Pegeldifferenz-Stereophonie. Warum kann die Samplingrate einen Einfluss haben.
- 8) Vergleiche beide Panning-Verfahren mit der Binauralsynthese.

Binauralsynthese mit dem SoundScape Renderer

Bisher haben wir nur statische Szenarien betrachtet. Zum Schluss soll es um Echtzeitanwendungen und Anbindung an eine DAW gehen. Hierzu werden wir das frei verfügbare, an der TU Berlin entwickelte Tool SoundScape Renderer (SSR) verwenden. Dieser läuft unter Mac OS und Linux und ist hier erhältlich: <http://spatialaudio.net/ssr/>.

- 9) Falls möglich installiere den SSR und startest seinen Binaural-Renderer.
- 10) Gehe auf <https://github.com/ssr-scenes/fh-koeln/tree/master/anechoic/audio> und lade die drei Audiodateien `BluesA_GitL_44100.wav`, `BluesA_GitR_44100.wav` und `BluesA_Voc_44100.wav` herunter. Öffne die Dateien im Binaural-Renderer des SSR. Sie sollten danach jeweils einzelne Punktquellen sein. Höre Dir das Stück an und bewege dabei die einzelnen Quellen. Stelle für eine der Quellen den Quelltyp auf ebene Welle um und achte auf den Unterschied, wenn Du die ebene Welle durch die Gegend bewegst.
- 11) Ordne die drei Quellen so an, dass es Dir gefällt. Speichere anschließend die Szene als `blues.asd` ab. Öffne die Datei und mache Dich mit dem Dateiformat vertraut. Erzeuge danach die Datei `stereo.asd`. Diese soll jetzt zwei Punktquellen beinhalten, die an die Positionen eines klassischen Stereoaufbaus gesetzt werden. Als Eingang sollen sie dieses mal keine wav-Dateien haben, sondern den Eintrag `<port>1</port>` und `<port>2</port>` enthalten, um anschließend in Jack eine DAW an diese Eingänge anschließen zu können (z.B. Ardour oder Bitwig haben einen Jack Ausgang).
- 12) Öffne die Datei `stereo.wav` mit einer DAW oder einem ähnlichem Programm und verbinde die beiden Kanäle zu den passenden Eingängen im SSR.

Binaural Room Scanning mit dem SoundScape Renderer

Im vorherigen Abschnitt haben wir den Binaural-Renderer des SSR verwendet. Dieser ist auf die Verwendung von HRIRs beschränkt. Im Folgenden wollen wir nun der Frage nachgehen, wie ein als BRIR Datensatz gemessenes Studio-Lautsprecher-Layout die HRIRs ersetzen kann. Hierzu benötigen wir den sogenannten *Binaural Room Scanning* (BRS) Renderer. Dabei wird dem SSR pro Lautsprecher eine 720-kanalige wav-Datei übergeben, die für alle möglichen Kopforientierungen in 1° Schritten die passenden Impulsantworten enthält.

- 13) Zunächst müssen wir aus dem gemessenen `BRIRs.sofa` Datensatz die passenden wav-Dateien für jeden Lautsprecher erstellen. Praktischerweise lässt sich dies auch mit der SFS Toolbox realisieren. Schaue Dir dazu die Hilfe der Funktion `ssr_brs_point_source()` an.
- 14) Wenn die einzelnen wav-Dateien erzeugt worden sind, muss eine asd-Datei mit der passenden Szene angelegt werden. Ein einzelner Lautsprecher Eintrag sieht dabei wie folgt aus:

```
<source name="L" model="point" properties_file="brs_loudspeaker1.wav">
  <port>1</port>
  <position x="-1.0" y="1.73"/>
</source>
```
- 15) Zum Schluss öffne die Datei `surround.wav` in einer DAW oder ähnlichem und verbinde ihre Kanäle zu den passenden Lautsprechern im SSR.