

# SweetSpotter

Xiangqiong Zeng

Xin Yang

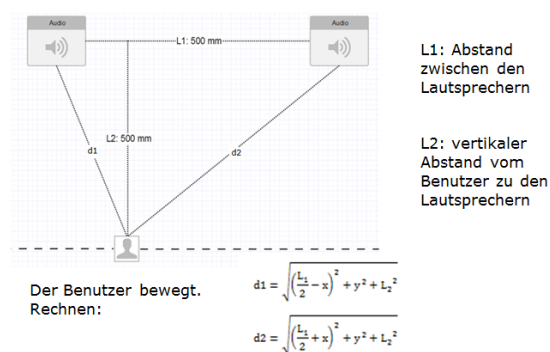
## 1. Einführung

Wenn wir mit zwei Lautsprechern Musik hören und sich dazwischen etwa auf der Symmetrieachse befinden, erleben wir eine sehr gute akustische Wahrnehmung, und diesen optimalen Bereich heißt man Sweet Spot. Aber das Problem ist, wenn wir sich außer diesem Bereich z.B. nach rechts bewegen, bewegt sich das Stereobild auch nach rechts und schließlich empfinden wir die Musik nur von dem rechten Lautsprecher. Es ist eine psychoakustische Auswirkung. Der akustische Grund dafür ist, die Musik kommt aus der rechten Seite wegen des Laufzeit- und Pegelunterschieds lauter und früher. Der Benutzer ist im konventionellen Stereo-System beschränkt. Aber wir können es verbessern. Das Ziel von unserer Projektarbeit --Sweetspotter ist, dass die beiden Lautsprechersignale je nach der Position des Zuhörers in Echtzeit anzupassen, damit das sogenannte Sweet Spot über dem gesamten Hörbereich erreicht und der Benutzer nicht mehr beschränkt ist.

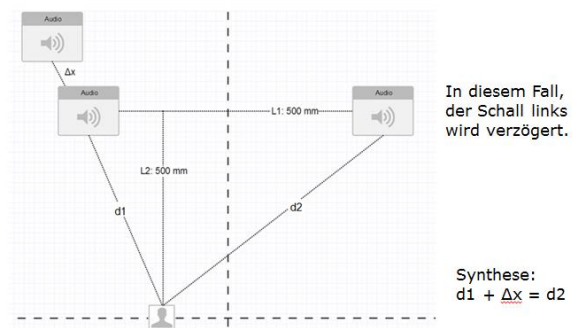
## 2. Theorie

Um den Laufzeitunterschied zu bekommen, müssen wir zuerst die Laufwege zu den beiden Lautsprechern rechnen. Das Modell ist unten links. L1 ist der Abstand zwischen beiden Lautsprechern, L2 ist der vertikale Abstand vom Zuhörer zu den Lautsprechern. Die beiden werden fixiert, und vor der Programmierung laut den eventuellen Lagen konfiguriert. Der Benutzer bewegt sich auf links-rechts Linie und die Laufwege d1 und d2 können dann laut dem Prinzip des Dreiecks berechnet werden.

Laufwege



Synthese des Modells

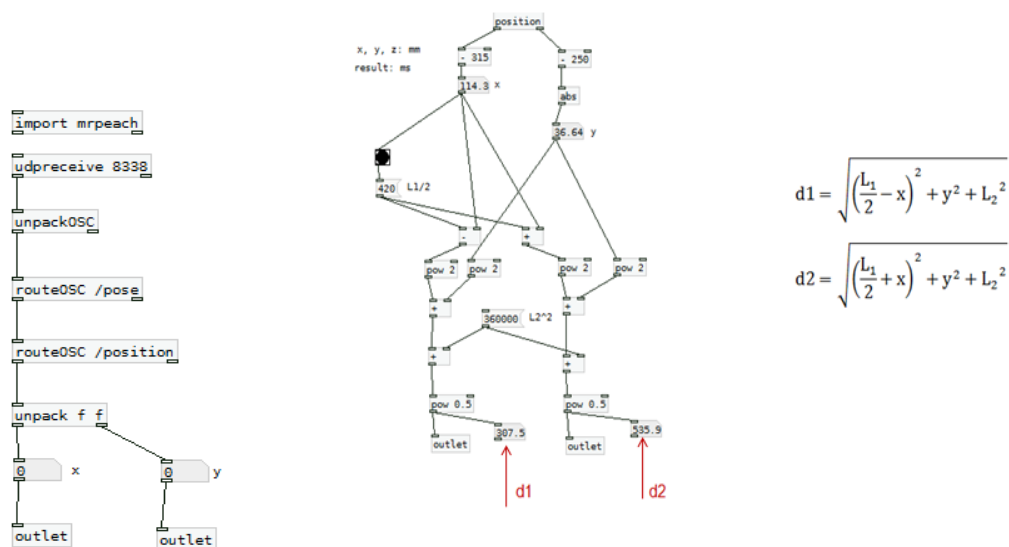


Jetzt machen wir eine Synthese, damit  $d1 + \Delta x = d2$  ist. In diesem Fall kann man das Musiksinal von der linken Seite um  $\Delta x$  durch die Schallgeschwindigkeit

verzögern, damit man die beiden Schallsignale gleichzeitig wahrnimmt, und so wird das Sweet spot hier theoretisch realisiert.

### 3. Umsetzung

Zuerst brauchen wir die Position von dem Benutzer. Für das Gesicht-tracking benutzen wir hier die Software FaceOSC und eine Webkamera. Face OSC erkennt das Gesicht mit vielen Parametern. Wir benutzen nur die Position vom Gesicht.



Oben links ist das Programm in pure data für FaceOSC. Dadurch können die Informationen über die Position erhalten werden. Und x ist für links-rechts-achse und y ist für oben-unten-achse. Oben rechts ist die Rechnung in Pure Data für die Laufwege entspricht den beiden Formeln.

### 4. Installieren

Die Software werden benötigt: PureData und FaceOSC.

Virtual Cable ist hilfreich.

### 5. Verwendungshinweis

Wenn man PureData und FaceOSC installiert hat, kann man jetzt diese Anwendung benutzen.

Zuerst öffnet man FaceOSC, dann „read audio.pd“. In dem selben Ordner soll es „t.wav“ Datei gibt. Im Datei „position.pd“ kann man die Parameter L1 und L2 ändern. Achtung: in den Eingabefelder sind  $\frac{L1}{2}$  und  $L2^2$ . Bitte nicht vergessen, DSP von Pd zu öffnen. Danach auf den Bang (nicht auf „start“) klicken und alles geht los. Um die Musik zu beenden clickt man auf „stop“.

Alternative kann man mit Virtual Cable diese Anwendung benutzen, es ist viel günstiger. Nach der Installierung soll man die Wiedergabe und Aufnahme auf „Virtual In/Output“ setzen, und in PureData bei Audioeinstellung Virtual Input als Input und Lautsprecher als Output setzen. So kann man alles mit „Sweetspotter“ hören. Statt „read audio.pd“ wird dann „read audio with Virtual Cable.pd“ benutzt.

Weitere Hinweise siehe Kommentar in den Pd-Dateien.