

TECHNISCHE UNIVERSITÄT DRESDEN

FAKULTÄT ELEKTROTECHNIK
UND INFORMATIONSTECHNIK

Virtuelle Realität

Anne Schröder

Maria Enge

Philipp Weiß

1. Einführung

Die Aufgabe beinhaltete die knackfreie Umsetzung des Sweet Spotters in Echtzeit und mit Positionsbestimmung durch Facetracking per Webcam.

Unter einem „Sweet Spotter“ versteht man in diesem Fall die Anpassung des Sweet Spots eines Lautsprechersystems, auf die jeweilige Position des Hörers.

Optional sollte ein Plugin für jede Audioanwendung integriert werden.

Die Implementierung erfolgte in der grafischen Entwicklungsumgebung Max/MSP (32-bit) und das Facetracking konnte durch das in Max/MSP integrierte Videobearbeitungs-Programm Jitter realisiert werden.

Zuletzt ermöglichte der Audio-Treiber VB-Cable jedes Signal anderer Audioanwendungen in das Programm zu übertragen.

Um ohne Max/MSP auszukommen, besteht die Idee den Patch als alleinstehendes, ausführbares Programm zu kompilieren.

2. Vorüberlegungen

Die Wahl fiel auf Max/MSP, da bei Pure Data nur begrenzte Möglichkeiten des Facetrackings (faceAPI, FaceOSC, Jitter) zur Verfügung stehen, die nicht in die Software integrierbar sind. Was noch für Max/MSP spricht, ist die Möglichkeit ein Audio-Plugin (Jamoma/Plugtastic) zu installieren.

Zudem entschied sich die Projektgruppe dafür, die Positionsbestimmung per Eyetracking zu realisieren, da der Augenabstand bei jedem Menschen nahezu gleich ist und somit außer der seitlichen Mittenabweichung, ohne Weiteres auch der Abstand vom Lautsprecher-Setup berechnet werden kann.

Um globales Puffern zu vermeiden, soll einseitig lokal verzögert und der andere Kanal in Echtzeit durchgegeben werden.

Anstatt ein Audio-Plugin zu entwerfen, welches in Anwendungen eingebunden werden muss, findet in Windows VB-Cable Verwendung, das den Audiomix ins Programm durchgibt.

3. Installation

Für das Programm ist die Installation folgender Anwendungen erforderlich:

- Max/MSP (32-bit, da somit Jitter integriert)
- cv.jit collection
- VB-Cable (Audio Device Driver)

Für Mac-Benutzer ist darüber hinaus zu beachten das „dx“ in *jit.dx.grab* durch „qt“ zu ersetzen, da hier statt DirectX das Programm QuickTime genutzt wird.

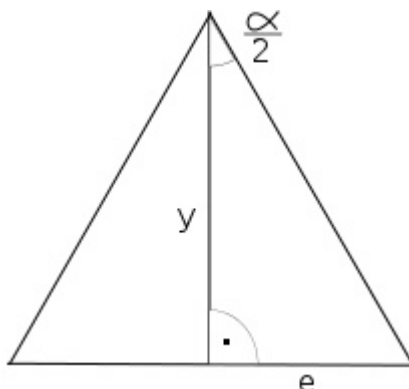
4. Berechnungen

Für die Realisierung des Sweet Spots gilt es einige Berechnungen vorzunehmen.

Diese betreffen einerseits die Laufzeitdifferenz und andererseits die Pegeldifferenz des Audio-Signals, sowie deren Aufteilung auf die Lautsprecher.

Mit einer Fallunterscheidung gelingt es einen flexiblen Sweet Spot rechts bzw. links der Hauptsache zu erzeugen.

Die Positionsbestimmung erfordert eine Berechnung der Kamera-Daten in die Positionsdaten, die die Grundlage für die weiteren Berechnungen bildet:



α ... Kamerawinkel

d ... von Kamera erfasster Augenabstand [px]

0.065 m ... realer Augenabstand beim Menschen

80 ... Entsprechung von e in Pixeln

$$\frac{0.065 \text{ m}}{e} = \frac{d}{80} \quad \rightarrow \quad e = \frac{0.065 \text{ m} \cdot 80}{d}$$

$$y = \frac{e}{\tan(\frac{\alpha}{2})} \rightarrow y = \frac{0.065 \text{ m} \cdot 80}{d \cdot \tan(\frac{\alpha}{2})}$$

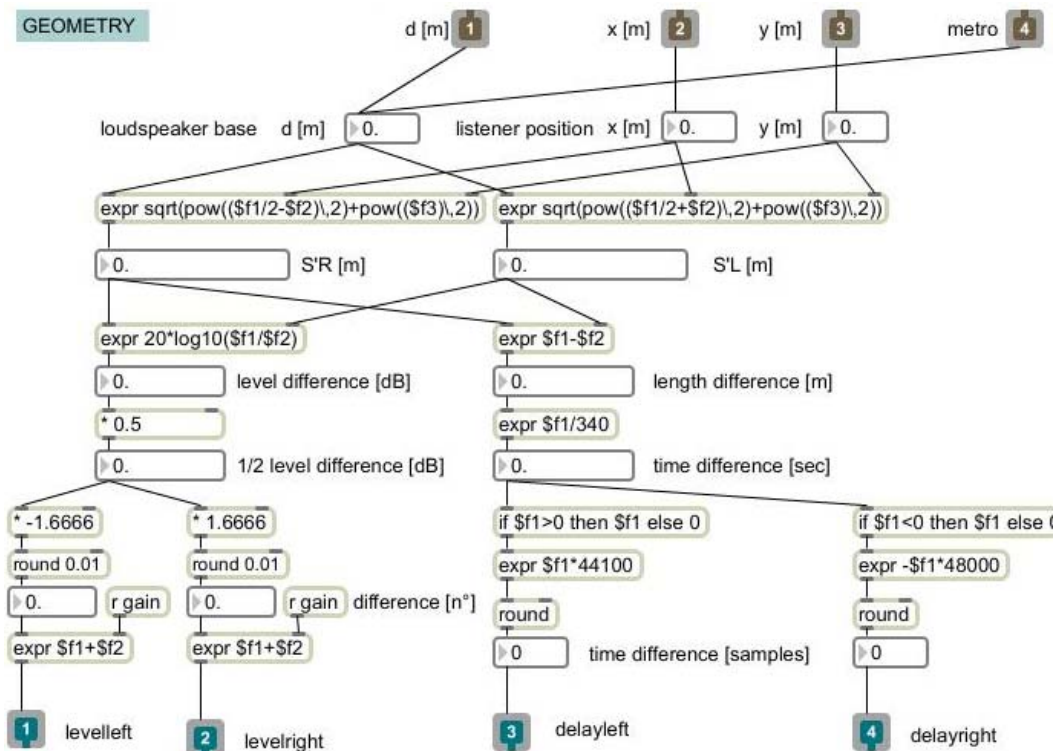
Der Umrechnungsfaktor k von Pixeln in Metern ergibt sich somit zu:

$$k = \frac{y \cdot \tan(\frac{\alpha}{2})}{80}$$

5. Umsetzung als Programm

Die Nachregelung der Laufzeit wird im Programm über eine „if“-Abfrage und Delays realisiert, da ansonsten global gepuffert werden müsste.

Die Dämpfung bzw. Verstärkung des Pegels ist für beide Lautsprecher realisiert, um die Schallleistung aufrecht zu erhalten, hat aber den Nachteil des möglichen Übersteuerns (ausgeglichen durch gain-Regler).



6. Hinweise für die Anwendung

Die Auflösung des Kamera-Bildes ist mit 160x120 festgelegt (s. u.a. `cv.jit.resize @size 160 120`).

Hier sind keine Veränderungen vorzunehmen, da mit dieser Pixel-Anzahl nicht-variabel weiter gerechnet wird.

Weiterhin wird in den Berechnungen von einem Augenabstand von 6,5 cm ausgegangen.

An dieser Stelle sollte ebenfalls keine Modifikation vorgenommen werden, da mit diesem Wert fest iteriert wird.

Das Kamera-Bild ist als Spiegel (*mirror*) einzustellen, falls nicht bereits so initialisiert.

Der Kamera-Öffnungswinkel (*cam fov*) kann je nach Modell der Webcam im Underpatch des Eyetrackings festgelegt werden.

Er ist auf 60° voreingestellt, was für die meisten integrierten Webcams ein angemessener Wert ist.

Die Abtastrate ist mit 48 kHz voreingestellt, kann aber nach Belieben in p Geometry anders festgelegt werden.

Der Kopf kann geneigt, sollte aber nicht verdreht werden, da sonst das Eyetracking seine Aufgabe nicht zufriedenstellend erfüllen kann und der Höreindruck sowieso verfälscht würde.

Im Hauptfenster können im *INTERFACE* grundsätzliche Einstellungen vorgenommen und eingesehen werden.

