

Schärfung der Richtungslokalisation mittels Directional Audio Coding für Higher Order Ambisonics

Bachelorarbeit aus Aufnahmetechnik 1, SE

Julian
Koch

Betreuung: Dr. Alois Sontacchi

Graz, 10. August 2016



institut für elektronische musik und akustik



Zusammenfassung

DirAC¹ hat sich als gängige Methode zur Richtungsschärfung etabliert. Bis jetzt wurde es aber vor allem für VBAP² beschrieben, bei dem die geometrische Anordnung der Lautsprecher schon im Vorhinein bekannt sein muss. In dieser Arbeit wurde das DirAC Verfahren für Higher Order Ambisonic umgesetzt, welches neben anderen Vorteilen auch die Bearbeitung ohne Kenntnis der Lautsprecheranordnung ermöglicht. Außerdem wurde zur Frequenzanalyse die CQT³ verwendet, welche eine gehörangepasste Frequenzanalyse ermöglicht.

-
1. Directional Audio Coding
 2. Vector Based Amplitude Panning
 3. Constant-Q-Transformation

Abstract

DirAC¹ is an established method to improve the perceived direction of arrival. Up to now this method has been particularly described for VBAP². The drawback of this method is that the positions of the loudspeakers must be previously known. In this thesis the DirAC method has been described for Higher Order Ambisonics, which has multiple advantages, one of them being that the DirAC algorithm can also be used without any knowledge about the loudspeaker configuration. Furthermore, a frequency analysis method, that is adapted to the human auditory system, has been used (CQT³).

-
1. Directional Audio Coding
 2. Vector Based Amplitude Panning
 3. Constant-Q-Transformation

Inhaltsverzeichnis

1	Einleitung	5
2	Ambisonics B-Format und theoretische Grundlagen	5
2.1	Aufnahme im B-Format	6
2.1.1	Soundfield Mikrofon	7
2.2	Kugelkoordinatensystem	8
3	DirAC	8
3.1	Grundlegendes	8
3.2	Richtungsanalyse	10
3.2.1	Glättung des Richtungsvektors	11
3.3	Diffusennessanalyse	13
3.4	Verwendung virtueller Mikrofone	13
3.5	Probleme von DirAC	14
4	Frequenzanalyse	14
4.1	Constant-Q-Transformation	15
5	Realisierung des MatLab Programms	16
5.1	CQT Implementierung	16
5.2	Signalfluss	16
5.3	Funktionen innerhalb des MatLab Programms	19
5.4	Die Bedienoberfläche des Programms	20
6	Binaurale Wiedergabe in Reaper	21
7	Schlussfolgerung und Ausblick	23

1 Einleitung

Ambisonics ist ein 3D Soundproduction System, dessen Qualität der Richtungslokalisation und Breite des Sweet Spots von der Ambisonic Ordnung abhängig sind (vgl [FZS15]). Jedoch ist das direkte Aufnehmen in höheren Ordnungen sehr aufwändig. Man benötigt dafür sehr spezielle Richtcharakteristiken, welche teils nur durch viele Mikrofonkapseln erreicht werden können (vgl [IS11]). Eine alternative Variante dazu ist es, sich bei der Aufnahme auf eine niedrigere Ordnung zu beschränken, welche leichter aufzunehmen ist. Danach kann das aufgenommene Signal mittels Directional Audio Coding bearbeitet werden. DirAC erlaubt es relevante Informationen aus dem B-Format erster Ordnung zu extrahieren, um das Signal anschließend in höherer Ordnung zu resynthetisieren. Vgl dazu [FZS15]

Das Hauptaugenmerk liegt in dieser Bachelorarbeit auf der Implementierung des DirAC Verfahrens für Ambisonics in ein MatLab Programm, wobei eine gehörangepasste Methode zur Frequenzanalyse verwendet wird (CQT(siehe Abschnitt 4.1)). Dieses MatLab Programm liest Dateien im B-Format erster Ordnung (Kanäle W,X,Y und Z) ein und gibt anschließend ein Multikanal-File der gewünschten Ordnung aus. Ausgangspunkt dafür ist ein am IEM Graz erstelltes MatLab Skript von Franz Zotter, welches in folgenden Punkten verbessert werden soll:

- Unterscheidung zwischen direktionalen und diffusen Signalanteilen und Resynthese des Direkt- und Diffusanteils mittels passenden Methoden
- Verwenden einer gehörangepassten Frequenzanalyse (CQT)
- Erstellen einer Benutzeroberfläche für die leichtere Bedienung des Programms, speziell für Musiker

Ergänzend werden in dieser Bachelorarbeit die theoretischen Aspekte von DirAC und den anderen implementierten Funktionen behandelt.

2 Ambisonics B-Format und theoretische Grundlagen

In diesem Abschnitt wird das, für diese Arbeit essenzielle B-Format und dessen Aufnahme näher betrachtet.

Die generelle Idee hinter Ambisonics ist das räumliche Aufteilen des 3-dimensionalen Schallfelds in mehrere Funktionen, die sogenannten Kugelflächenfunktionen [FZS15]. Liegen die Kugelflächenfunktionen des Schallfelds direkt vor, so spricht man vom soge-

nannten B-Format. Abbildung 1 zeigt die Kugelflächenfunktionen bis zur 3ten Ordnung.

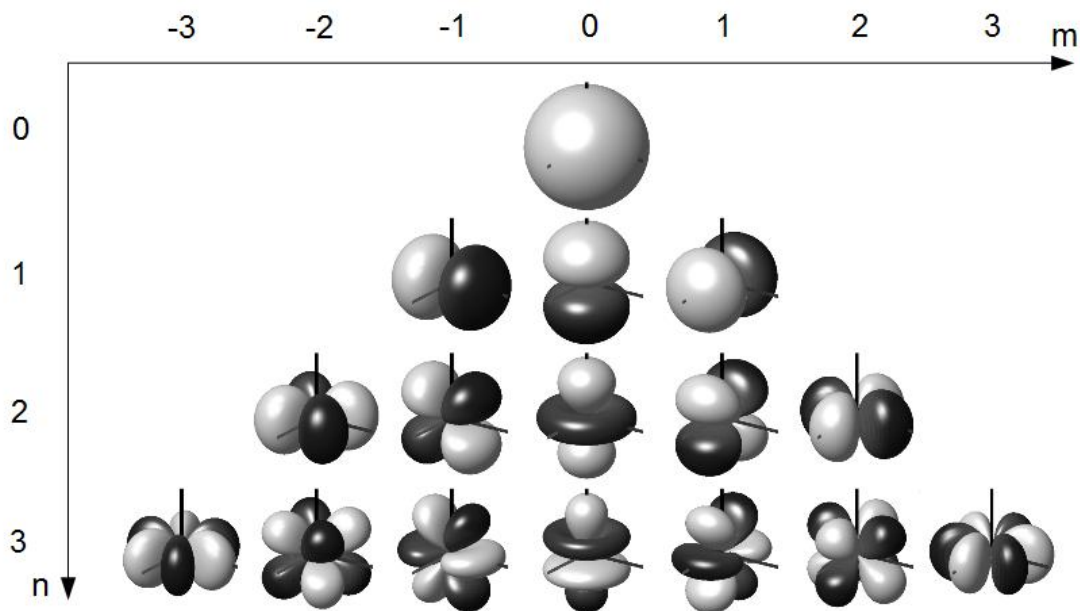


Abbildung 1: Kugelflächenfunktionen einschließlich 3. Ordnung, wobei n die Ordnung und m den Grad bezeichnet. Die Kugelflächenfunktion 0. Ordnung entspricht Richtcharakteristik Kugel, jene 1. Ordnung entsprechen Richtcharakteristik Acht/ Quelle: Wikipedia⁴

Höhere Ordnungen von Ambisonics enthalten ebenfalls die Kugelflächenfunktionen der niedrigeren Ordnungen. Die Anzahl der benötigten Gesamtkanäle für eine bestimmte Ambisonics Ordnung N lässt sich also leicht mit Hilfe der folgenden Formel berechnen

$$K = (N + 1)^2 \quad (1)$$

Die Ambisonics Kanäle beschreiben das komplette Schallfeld und sind unabhängig von der Lautsprecheranordnung bei der Wiedergabe. Um die Lautsprecher signale zu erhalten, werden die Ambisonics Kanäle durch Multiplikation mit einer Decodermatrix für jeden Lautsprecher gewichtet und addiert.

2.1 Aufnahme im B-Format

Somit enthält das B-Format erster Ordnung vier Kanäle. Diese vier Kanäle entsprechen den Kugelflächenfunktionen erster und nullter Ordnung. Aufgrund der relativ simplen Form dieser Funktionen ist die Aufnahme dieser Kanäle mittels Mikrofonen verfügbarer Richtcharakteristik realisierbar.

4. https://en.wikipedia.org/wiki/Ambisonics#/media/File:Spherical_Harmonics_deg3.png
 Autor: Franz Zotter/ abgerufen am 27.6.2016

Für die direkte Aufnahme im B-Format erster Ordnung werden also vier Mikrofone benötigt: Drei Druckgradienten Mikrofone (Richtcharakteristik Acht), welche in Richtung der Raumachsen ausgerichtet werden und ein Druckmikrofon (Richtcharakteristik Kugel). Für die direkte Aufnahme der Ambisonic Kanäle mit hoher Ordnung würde man Richtcharakteristiken benötigen, welche nicht bzw. nur schwer realisierbar sind.

2.1.1 Soundfield Mikrofon

Wie bereits erwähnt ist die direkte Aufnahme im B-Format 1. Ordnung mit gebräuchlichen Mikrofonen möglich. Es existieren aber auch fertige Anordnungen von Mikrofonkapseln, welche durch eine günstigere geometrische Anordnung Richtungsfehler minimieren können. Eine dieser fixen Mikrofonanordnungen ist das Soundfield Mikrofon (nähere Infos auf der Herstellerwebsite⁵). Bei diesem Mikrofon werden vier Mikrofonkapseln mit Nierencharakteristik in der Form eines Tetraeders angeordnet. Als Ergebnis erhält man das sogenannte A-Format. Um dieses in das B-Format umzuwandeln, werden die Zeitsignale der Nierenmikrofone kombiniert:

$$\begin{aligned} w &= LF + RF + LB + RB \\ x &= LF + RF - (LB + RB) \\ y &= RF + RB - (LF + LB) \\ z &= RF + LB - (LF + RB) \end{aligned} \tag{2}$$



Abbildung 2: Soundfield Mikrofon zur Aufnahme im A-Format, welches anschließend auf das B-Format transformiert wird⁶

wobei die Zeitsignale der Einzelmikrofone der Soundfieldmikrofonanordnung wie folgend bezeichnet wurden

5. <http://www.tslproducts.com/soundfield/soundfield-sps200-software-controlled-microphone/>

6. Quelle: <http://www.tslproducts.com/wp-content/uploads/2013/05/sps-software-mic-thumb.png/> abgerufen am 27.6.2016

LF...Left Front
 RF...Right Front
 LB...Left Back
 RB...Right Back

Vgl dazu [IS11], [BC05]

Da sich bei der Aufnahme die Kapseln aller Mikrofone theoretisch an gleicher Stelle befinden müssten und dies in der Praxis nicht möglich ist, treten Abbildungsfehler auf. Auf die Auswirkung dieser Abbildungsfehler wird in dieser Arbeit nicht weiter eingegangen, jedoch ist es wichtig zu wissen, dass Fehler auftreten. Diese Erkenntnis wird später noch zur Verringerung der Rechenleistung genutzt. Weitere Informationen zum Richtungsfehler sind unter [BC05] und [IS11] zu finden.

2.2 Kugelkoordinatensystem

Um jeden Punkt im dreidimensionalen Raum mittels Kugelkoordinaten beschreiben zu können, werden 3 Parameter benötigt: der Azimutalwinkel (φ), der Polarwinkel (θ) und der Abstand zum Koordinatenursprung (r). Da für die Anwendung dieser Arbeit nur die Richtung von Bedeutung ist, reichen hier φ und θ . Es wird jenes System verwendet, welches in der Mathematik bzw. Physik üblich ist. Hier gilt für die Umrechnung von kartesischen Koordinaten auf Kugelkoordinaten:

$$\theta = \arccos \frac{z}{\sqrt{x^2 + y^2 + z^2}} \quad (3)$$

$$\varphi = \text{atan2}(y, x) = \begin{cases} \arctan\left(\frac{y}{x}\right) & x > 0 \\ \text{sgn}(y) \frac{\pi}{2} & x = 0 \\ \arctan\left(\frac{y}{x}\right) + \pi & x < 0 \wedge y \geq 0 \\ \arctan\left(\frac{y}{x}\right) - \pi & x < 0 \wedge y < 0 \end{cases} \quad (4)$$

3 DirAC

3.1 Grundlegendes

DirAC (kurz für Directional Audio Coding) ist eine Methode zur Synthese und Resynthese von Audiodateien mittels weniger Parameter, welche sich für den Höreindruck beim Menschen als relevant erwiesen haben. Die Grundidee für DirAC wurde von der „Spatial

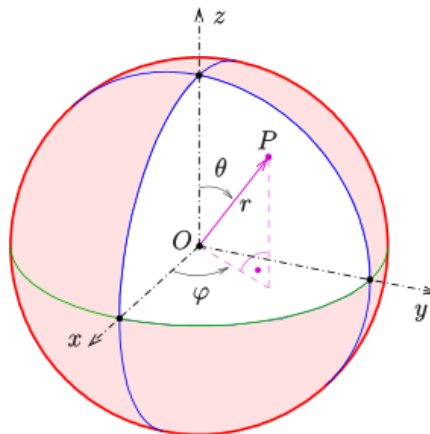


Abbildung 3: In der Mathematik bzw. Physik übliches Kugelkoordinatensystem zum Beschreiben der Richtung mit dem Azimutalwinkel φ und dem Polarwinkel θ , welches in dieser Arbeit verwendet wird/ Quelle: Wikipedia ⁷

Impulse Response Rendering-Technik“ (SIRR) übernommen (für weitere Infos zu SIRR wird auf [MP05], [PM06] und [Hos11] verwiesen).

Der DirAC Algorithmus funktioniert wie folgend: Als erstes wird das eingelesene Signal, welches im B-Format vorliegt sowohl zeit- als auch frequenztechnisch unterteilt und für jede dieser Unterteilungen jeweils die Richtung und die Diffusität des Signals berechnet: Durch Analyse des Richtungsvektors erhält man einen Azimutal- und einen Polarwinkel (φ bzw. θ), welche die Richtung eines Schallereignisses in Kugelkoordinaten beschreiben, sowie auch einen Parameter zur Beschreibung der enthaltenen Diffusität im Signal (Ψ).

Anschließend erfolgt die Resynthese des ursprünglichen Signals aus den zuvor gewonnenen Informationen. Dazu werden zwei verschiedene Techniken benötigt, eine zur Erzeugung des Direkt- und eine zur Erzeugung des Diffusanteils.

Für die Synthese des Direktanteils werden virtuelle Punktquellen verwendet. Üblicherweise wird für diese Realisierung VBAP verwendet. Dafür muss jedoch die endgültige Lautsprecherkonfiguration schon zuvor bekannt sein. In dieser Arbeit wird stattdessen das Ambisonics B-Format verwendet, welches unabhängig von der Lautsprecherkonfiguration funktioniert. Dies wird realisiert, indem die Kugelflächenfunktionen in der Richtung des erhaltenen Richtungsvektors ausgewertet werden, mit Hilfe der Decodermatrix einer

7. <https://de.wikipedia.org/wiki/Kugelkoordinaten#/media/File:Kugelkoord-def.svg/>
abgerufen am 27.6.2016

idealen Lautsprecheranordnung auf die Lautsprecherkoeffizienten transformiert werden und anschließend ein Signal durch Multiplikation mit den Lautsprecherkoeffizienten in jene Richtung „gepanned“ wird. Im einfachsten Fall wird als Signal der W-Kanal (omnidirektional) herangezogen. Jedoch besteht auch die Möglichkeit durch Kombinationen von W-, X-, Y- und Z-Kanal andere Richtcharakteristiken zu erreichen, um den Raumanteil im Direktsignal zu minimieren und die Richtungsinformation im Diffusteil zu bewahren (genauer dazu in Abschnitt 3.4).

Zur Reproduktion der Diffusanteile gibt es mehrere Ansätze. So werden in [LKDP11] frequenzabhängige Verzögerungen, welche konstant über die Zeit, aber für jeden Lautsprecher unterschiedlich sind vorgeschlagen. In [Pul07] hingegen wird das Signal durch Faltung mit exponentiell abklingendem weißen Rauschen, welches für jeden Lautsprecher unterschiedlich sein muss dekorreliert. Im Zuge dieser Bachelorarbeit wurden die oben genannten Methoden der Diffusitäts-Reproduktion implementiert und getestet. In der Literatur befasst sich DirAC meistens mit VBAP, wo ein Ansteuern der einzelnen Lautsprecher möglich ist. Da für die Erzeugung eines diffusen Eindrucks das Signal für jeden einzelnen Lautsprecher unterschiedlich modifiziert wird und da bei Ambisonic nie nur ein einziger Lautsprecher einen Beitrag zum Schallfeld leistet, funktionieren hier nicht unbedingt die oben vorgeschlagenen Methoden am besten. Außerdem sind auf Grund des hohen Rechenaufwands nicht alle Methoden (so wie die in [Pul07] vorgeschlagene Methode) für diese DirAC Implementierung geeignet. Die schlussendlich verwendete Methode erzeugt den diffusen Eindruck dadurch, dass die Phase in jedem CQT Fenster, sowohl zeit- und frequenztechnisch, als auch für jeden Lautsprecher zufällig gesetzt wird (diese Methode wird in [Pul07] auch kurz erwähnt). Die Vorteile dieser Methode sind die leichte Implementierung und der geringe Rechenaufwand, jedoch kann es bei dieser Methode auch zu Klangfärbungen kommen.

Der Direkt- und Diffusennessanteil werden anschließend dem Diffusenness-Parameters entsprechend zusammengemischt. Da es sich hier um ein Energiemaß handelt wird der Diffusennessanteil mit $\sqrt{\Psi}$ multipliziert und der Direktanteil mit $\sqrt{1 - \Psi}$ multipliziert und die beiden gewichteten Anteile anschließend addiert (vgl [Pul07]).

3.2 Richtungsanalyse

Durch Berechnung des Intensitätsvektors (und der daraus abgeleiteten Richtungsinformation) lässt sich die Richtung des Schallereignisses bestimmen. Das Ausgangssignal der Druckgradienten-Mikrofone ist proportional zur Schallschnelle der jeweiligen Richtung, während das Ausgangssignal des Druckmikrofons proportional dem Schalldruck am Ort der Mikrofonanordnung ist. Der Richtungsvektor (**D**) zeigt in die entgegengesetzte Richtung wie der Intensitätsvektor und berechnet sich zu

$$\mathbf{D}(k, n) = \text{Re}\{W^*(k, n)\mathbf{V}(k, n)\} \quad (5)$$

Siehe [FP06, 5]

wobei hier

k ...Frequenzindex

n ...Zeitindex

W^* ...konjugiert komplexe CQT des Schalldrucks

\mathbf{V} ...CQT des Schallschnellevektors ($\mathbf{v}(n) = x(n)e_x + y(n)e_y + z(n)e_z$)

Nun kann der Richtungsvektor mittels Gleichung 3 und 4 in Kugelkoordinaten umgewandelt werden. So erhält man die relevanten Winkel zur Beschreibung der Richtung.

3.2.1 Glättung des Richtungsvektors

Zur Vermeidung von sprunghaften Änderungen des Richtungsvektors ist eine Glättung über die Zeit notwendig. Ohne Glättung des Richtungsvektors kann es auf Grund der sehr kurzen Analysefenster der CQT bei hohen Frequenzen zu sehr schnellen Richtungsänderungen kommen, welche Artefakte verursachen (Clicks, Flirren etc.). Glättet man jedoch den Verlauf des Richtungsvektors direkt, entstehen Richtungen, die im ursprünglichen Signal eigentlich nicht vorhanden waren. Als Alternative bietet sich eine Glättung der Lautsprecher-Gewichtungs-Koeffizienten an, welche angeben wie stark der Beitrag eines einzelnen Lautsprechers am Schallfeld ist. Somit wird eine sprunghafte Änderung an einem Lautsprecher erst langsam eingefadet und verhindert so Artefakte. Da die Richtungsinformation momentan in Form der Sphärisch Harmonischen vorliegt, ist eine Transformation in den Ortsbereich (Lautsprecher-Gewichtungs-Koeffizienten) notwendig.

Um in den Ortsbereich zu gelangen, wird die Ambisonics Decoder-Matrix einer idealen⁸ Lautsprecheranordnung verwendet. Diese Decoder-Matrix erhält man, indem man die Sphärisch Harmonischen an den Positionen der Lautsprecher dieser Anordnung auswertet. Danach wird das Signal (welches als Sphärisch Harmonische vorliegt) durch Matrixmultiplikation mit der Decoder-Matrix auf die Lautsprecher projiziert. Als Beispiel folgt für erste Ordnung eine Transformation der Sphärisch Harmonischen Richtungsinformation (W_D, X_D, Y_D, Z_D) mittels Matrixmultiplikation mit der Decodermatrix der Lautsprecheranordnung ($W_{Li}, X_{Li}, Y_{Li}, Z_{Li}$) auf die Lautsprecherkoeffizienten (g_i), wobei hier 3 Lautsprecher verwendet werden und i den i -ten Lautsprecher bezeichnet. Zur besseren Übersichtlichkeit werden hier für die Richtungsinformation keine Zeit- bzw. Frequenzindizes angegeben.

8. Die Lautsprecher sind gleichmäßig auf einer Kugeloberfläche verteilt

$$\begin{bmatrix} W_D & X_D & Y_D & Z_D \end{bmatrix} \times \begin{bmatrix} W_{L1} & W_{L2} & W_{L3} \\ X_{L1} & X_{L2} & X_{L3} \\ Y_{L1} & Y_{L2} & Y_{L3} \\ Z_{L1} & Z_{L2} & Z_{L3} \end{bmatrix} = \begin{bmatrix} g_1 & g_2 & g_3 \end{bmatrix} \quad (6)$$

Anschließend werden die Lautsprecher-Koeffizienten ($g_i(k, n)$) für jeden Lautsprecher i mit der Energie und Diffusität gewichtet und mithilfe eines rekursiven Tiefpass-Filters 1. Ordnung über die Zeit geglättet, wobei die Polstelle des Filter bei α liegt [LKDP11, Eq. 7].

$$\hat{g}_i(k, n) = \alpha \cdot E_I(k, n) \cdot \sqrt{1 - \Psi(k, n)} + (1 - \alpha) \cdot \hat{g}_i(k, n - 1) \quad (7)$$

Die momentane Energie, welche im Analysefenster vorhanden ist (E_I) lässt sich in der Frequenzdomäne mit folgender Formel berechnen ([LKDP11, Eq. 2])

$$E_I = \frac{|W|^2 + |\mathbf{V}|^2/2}{2} \quad (8)$$

Bei STFT⁹ Implementierung wäre ein α , welches sich für die verschiedenen Frequenzbins unterscheidet, wünschenswert, um den Eigenschaften des menschlichen Gehörs zu entsprechen. Für hohe Frequenzen ist eine schwächere Glättung erwünscht, als für tiefe Frequenzen. Da die CQT aber schon dem Gehör angepasst ist und somit für tiefe Frequenzen weniger Frequenzbins zur Verfügung stehen, als für hohe Frequenzen, ist ein konstantes α zufriedenstellend. Für diese Arbeit wurde ein $\alpha = 0.15$ gewählt.

Anschließend müssen noch die Lautsprecher Koeffizienten normiert werden, um die Energie im Direktanteil zu erhalten. Es soll gelten

$$\sum_i \hat{g}_i^2(k, n) = 1 \quad (9)$$

Diese Methode der Glättung der Koeffizienten wird in [LKDP11, Eq. 10] vorgeschlagen. Jedoch wird durch diese Methode die Energie in den höheren Ordnungen verringert. Die Glättung klingt sauber, jedoch ist die Richtungsschärfung auf Grund der geringeren Energie in den höheren Ordnungen nicht mehr optimal.

Eine Alternative dazu wäre es, die Lautsprecherkoeffizienten ungewichtet mit demselben Filter wie zuvor zu glätten. Da jedoch bei dieser Methode nicht gewährleistet wird, dass die Energie im Direktanteil dieselbe bleibt, müssen die Koeffizienten noch im Gesamten mit dem gleichen Faktor skaliert werden.

9. Short-Time-Fourier-Transformation

3.3 Diffusenessanalyse

Ein wichtiger Punkt bei der Schärfung der Richtung durch Kodierung in höheren Ordnungen ist, dass durch die Verwendung von Punktquellen das gesamte Signal in eine Richtung projiziert wird und dadurch die diffusen Umgebungsgeräusche bzw. der Raumeindruck verloren geht. Das Resultat ist ein räumlich sehr präzises, aufgrund der fehlenden Rauminformationen aber künstlich klingendes Signal. Deswegen muss zusätzlich Diffusität erzeugt werden. Um das ursprüngliche Signal auf die enthaltene Diffusität zu untersuchen, wird der durch Berechnung erhaltene Intensitätsvektor auf die Gesamtenergie bezogen. Diese neu erhaltene Größe wird als Diffusitäts-Parameter (Ψ) bezeichnet. Die Werte von Ψ befinden sich zwischen 0 (keine Diffusität - ebene Welle aus nur einer Richtung) und 1 (nur diffuser Schall).

Ψ ist sowohl zeit- als auch frequenzabhängig.

$$\Psi = 1 - \frac{||\mathbf{I}||}{cE} \quad (10)$$

Für die Implementierung des Diffusitäts-Parameters wird bei der Verwendung der STFT bzw. CQT Methode folgende Formel verwendet [LKDP11, Eq 3]:

$$\Psi(k, n) = 1 - \frac{\sqrt{2} ||\text{Re}\{W^*(k, n)\mathbf{V}(k, n)\}||}{|W(k, n)|^2 + |\mathbf{V}(k, n)|^2/2} \quad (11)$$

3.4 Verwendung virtueller Mikrofone

Um die Trennung des Direkt- und Diffusanteils bestmöglich zu gewährleisten, ist es nötig, dass das Signal für den Direktanteil möglichst wenig Diffusität enthält. Dadurch, dass das Signal als Punktquelle in genau eine Richtung gelegt wird, verringert sich die wahrgenommene Diffusität schon. Durch Kombination von W-, X-, Y- und Z-Kanal können Richtcharakteristiken in die Richtung der virtuellen Lautsprecher gebildet werden, welche die Diffusität noch zusätzlich verringern.

Die Verwendung von Nierencharakteristik bringt für den Direktanteil eine Verringerung der Diffusität um durchschnittlich 4.8dB und wird durch folgende Kombination erreicht ([Pul07]):

$$M_i = \sqrt{2}W + X\cos(\phi_i)\sin(\theta_i) + Y\sin(\phi_i)\sin(\theta_i) + Z\cos(\theta_i) \quad (12)$$

wobei M_i das Signal des virtuellen Nieren-Mikrofons, welches auf den i -ten Lautsprecher gerichtet ist bezeichnet. ϕ_i und θ_i geben die Position der Lautsprecher in Kugelkoordinaten an.

Die Formel unterscheidet sich etwas von der in [Pul07] beschriebenen, da in diesem Paper ein anders definiertes Kugelkoordinatensystem verwendet wird.

Ebenso ist es sinnvoll diese Signale für die Erzeugung des Diffusanteils zu verwenden. Hier wird die Diffusität nicht verringert, da alle Lautsprecher gleich angesteuert werden. Durch die Verwendung der Richtcharakteristik in Richtung der Lautsprecher bleibt aber auch die Richtungsinformation im Diffusanteil erhalten.

3.5 Probleme von DirAC

Die DirAC Methode hat aber auch gewisse Nachteile. So kommt es zu Problemen, wenn zur gleichen Zeit im gleichen Frequenzbereich mehrere Schallereignisse aus unterschiedlicher Richtung parallel auftreten. Da DirAC nur einen Richtungsvektor pro Zeit- und Frequenzbereich berechnet, zeigt der resultierende Richtungsvektor in eine Richtung, die mit keiner der parallel auftretenden Schallereignisse korrespondiert. So wird bei der Resynthese ein Signal an einem Ort erzeugt, an dem ursprünglich keines vorhanden war.

Wie bereits in Abschnitt 3.2.1 behandelt, entsteht ein weiteres Problem durch das Unterteilen des ursprünglichen Signals in Zeit- und Frequenzabschnitte. Der Richtungsvektor soll sich nicht sprunghaft ändern, da es ansonsten zu Artefakten wie "Clicks" kommen kann.

4 Frequenzanalyse

Wie bereits zuvor erwähnt, ist es notwendig das Ausgangssignal in Frequenzbänder zu unterteilen. Hierzu existieren mehrere Möglichkeiten. Die erste Möglichkeit ist, per Filterbank einzelne Frequenzbereiche herauszufiltern. Diese Methode erlaubt eine selbst gewählte Verteilung der Center-Frequenzen der Filter. Optimal werden die Center-Frequenzen an das menschliche Gehör angepasst. Somit liefert diese Methode eine hohe Qualität, bei jedoch hohem Rechenaufwand.

Die zweite Möglichkeit ist die Verwendung der STFT. Diese Methode hat den Vorteil, dass sie leicht implementierbar ist und vor allem für Echtzeitanwendungen (Telekonferenzen) geeignet ist, da der Rechenaufwand gering ist. Jedoch ergibt sich der Nachteil, dass die Frequenzbins der STFT linear über die Frequenzachse verteilt sind. Da jedoch eine Verdopplung der Frequenz als ein Tonhöhenunterschied von einer Oktave wahrgenommen wird, ist eine hohe frequenztechnische Auflösung im tiefen Frequenzbereich und eine niedrigere Auflösung im hohen Frequenzbereich vom Vorteil.

Dies ist mittels der CQT (Constant Q-Transformation) realisierbar. Nebst des Vorteils der geometrisch verteilten Frequenzbins (Formel 15), bringt die CQT auch einen anderen Vorteil mit sich. So wird durch die kürzere Fensterlänge für hohe Frequenzen die Zeitauf-

lösung verbessert, während für tiefe Frequenzen lange Fenster verwendet werden (vgl dazu [SKHD14]).

4.1 Constant-Q-Transformation

Die CQT erhält man durch Modifikation der STFT. Hierbei werden, wie bereits zuvor erwähnt, die Frequenzbins nicht linear wie bei der STFT gesetzt, sondern geometrisch verteilt (Gleichung 15 entsprechend).

Um eine generelle Formel für die CQT herzuleiten, geht man von der STFT aus.

$$X[k, i] = \sum_{n=0}^{N-1} x[n]w[n-i]e^{-j2\pi k \frac{n}{N}} \quad (13)$$

Bei der STFT liegen die Frequenzkoeffizienten mit konstantem Abstand bezüglich der Frequenzachse vor. Die Frequenzauflösung berechnet sich zu

$$\Delta f = \frac{f_s}{N} \quad (14)$$

f_sSamplingfrequenz

NAnzahl der Samples

iAnalysezeitpunkt des Fourierspektrums

kFrequenzindex

Gewünscht ist nun eine Verteilung der Frequenzbins, bei der der Abstand zwischen den einzelnen Koeffizienten einer geometrischen Folge entspricht:

$$f_k = f_0 \cdot 2^{\frac{k}{b}} \quad (15)$$

f_0Startfrequenz

bAnzahl der Frequenzbins pro Oktave

Aus Formel 14 folgt, dass bei konstanter Samplingfrequenz f_s die Frequenzauflösung von der Anzahl der verwendeten Samples abhängt.

Um nun eine variable Frequenzauflösung zu erhalten, wird also N variiert. Gewünscht ist eine bessere Frequenzauflösung für tiefe Frequenzen, welche zu hohen Frequenzen hin abnimmt. Dies wird durch eine frequenzabhängige Fensterfunktion (w_k) realisiert. Für tiefe Frequenzen werden längere Fenster verwendet, für höhere Frequenzen kürzere. Die Formel für die CQT lautet also

$$X[k, i] = \sum_{n=0}^{N-1} x[n] w_k[n - i] e^{-j2\pi k \frac{n}{N}} \quad (16)$$

5 Realisierung des MatLab Programms

5.1 CQT Implementierung

Zur Durchführung der CQT wurde die MatLab Toolbox von Christian Schörkhuber verwendet [SKHD14].

Dadurch, dass die Anzahl der Zeitbins von der höchsten Frequenz abhängt, entstehen für eine Grenzfrequenz von $fs/2$ enorme Datenmengen, deren Bearbeitung sehr rechenintensiv ist und somit lange dauert. Um dieses Problem zu umgehen, werden die Signale nur bis zu einer selbst wählbaren Grenzfrequenz analysiert und bearbeitet.

Dies ist nicht nur wichtig zur Verringerung der Rechenkomplexität, sondern in der Praxis auch aufgrund von Fehlern bei der Richtungslokalisierung mittels dem Intensitätsvektor für hohe Frequenzen nötig. Für Aufnahmen im B-Format mit dem Soundfield Mikrofon (welches vorwiegend als Quelle für dieses MatLab File verwendet werden soll) ergeben sich für Frequenzen über 3000Hz bereits Richtungsfehler, größer als 10° . Für höhere Frequenzen steigt der Richtungsfehler signifikant an, sodass die Schärfung der Richtung überflüssig wäre (vgl dazu [IS11, S.33]).

5.2 Signalfluss

Dieser Abschnitt beschreibt den in dieser Arbeit erstellten MatLab Code und dient vor allem dem besseren Verständnis des Codes, damit eine weitere Modifikation leichter fällt.

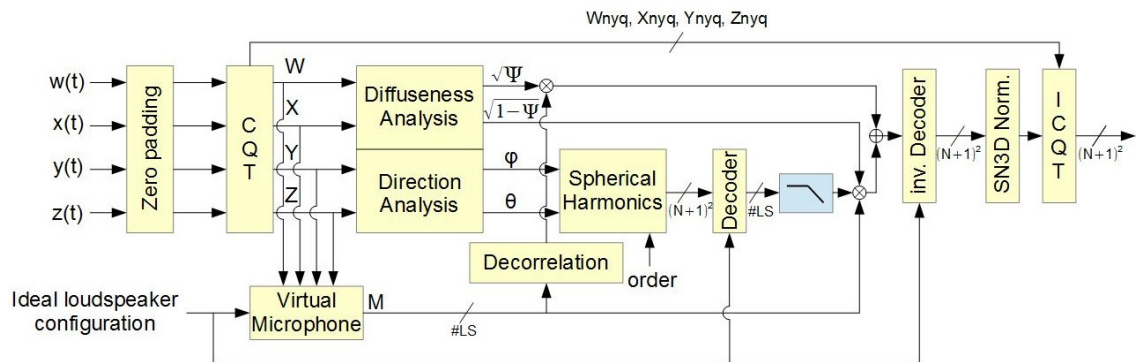


Abbildung 4: Blockdiagramm des MatLab Files für DirAC bei Ambisonic-Umsetzung

ANALYSE

Bevor mit der Bearbeitung begonnen wird, werden die einzelnen Kanäle noch im Zeitbereich in Bearbeitungsblöcke¹⁰ unterteilt und getrennt bearbeitet. Dies ist vor allem für lange Dateien notwendig, da ansonsten der Speicherbedarf sehr hoch wird.

Anschließend werden die eingelesenen Dateien noch vorne und hinten mit Nullen ergänzt. Dies ist für die CQT notwendig, da es ansonsten aufgrund der Filterung im Frequenzbereich zu Artefakten kommen kann. Die Zeitblöcke werden dann mittels CQT-Toolbox in den Frequenzbereich übergeführt. Da die CQT zusätzlich zum gewählten Frequenzbereich noch 2 zusätzliche Frequenzbins (unter und über den gewählten Grenzfrequenzen) zur perfekten Reproduktion erstellt, wird der Frequenzbereich über der oberen Grenzfrequenz zwischengespeichert, um später vor der Rücktransformation in den Zeitbereich wieder leicht zur Verfügung zu stehen. Der Bereich über der Grenzfrequenz wird nicht bearbeitet.

Als nächstes wird für jeden Lautsprecher einer virtuellen Lautsprecheranordnung ein eigenes Signal berechnet. Man kombiniert die Kanäle W,X,Y und Z so, dass eine Nierencharakteristik entsteht, welche in die Richtung des jeweiligen Lautsprechers zeigt (siehe Gleichung 12).

Aus den in den Frequenzbereich transformierten Kanälen W,X,Y und Z wird analog zu Gleichung 5 der Richtungsvektor berechnet. Der Richtungsvektor, welcher in kartesischen Koordinaten vorliegt, wird nun mittels Gleichung 3 und 4 in Kugelkoordinaten umgewandelt. Hierbei ist nur die Richtung des Vektors wichtig, der Betrag wird nicht weiter betrachtet.

Um die Diffusität zu berechnen, wird Gleichung 11 verwendet. Ebenfalls wird analog zu Gleichung 8 die Energie berechnet, welche in jedem Zeit- bzw. Frequenzbin vorhanden ist. Diese Information wird später bei der Glättung benötigt.

SYNTHESE

In der Synthesephase werden als erstes die Kugelflächenfunktionen der gewünschten Ordnung am analysierten Ort ausgewertet¹¹

10. Für das Programm wurde eine Länge von 5 Sekunden pro Block gewählt

11. Die Funktion „sh_matrix_real“, welche im MatLab Programm zum Auswerten der Sphärisch Harmonischen Verwendung findet, verwendet ein Kugelkoordinatensystem, in dem φ im Uhrzeigersinn läuft und nicht wie üblich gegen den Uhrzeigersinn. Daher ist es notwendig als Parameter $-\varphi$ zu übergeben.

Danach werden die Kanäle mittels Matrixmultiplikation mit der Decodermatrix einer idealen Lautsprecheranordnung auf die Lautsprecherkoeffizienten überführt (siehe Abschnitt 3.2.1). Diese werden dann mit einem rekursiven Tiefpass-Filter 1. Ordnung entsprechend Gleichung 7 über die Zeit geglättet. Wahlweise ist hier die Glättung mit oder ohne Gewichtung der Lautsprecherkoeffizienten durch die Energie und Diffusität möglich (Abschnitt 3.2.1).

Zur Erzeugung eines diffusen Signals wird anschließend für die resultierenden Lautsprechersignale die Phase zufällig zwischen $-\pi$ und π gesetzt. Dies geschieht für jeden Zeit- und Frequenzbin.

Darauf folgend wird das zuvor durch Gleichung 12 berechnete, lautsprecherabhängige Signal mit den Lautsprecherkoeffizienten multipliziert und ebenfalls der Diffusanteil dazugemischt. Da die Energie im Diffusanteil nicht erhalten bleibt und unter anderem auch von der Anzahl der virtuellen Lautsprecher abhängt, besteht in der Benutzeroberfläche die Möglichkeit den gesamten Pegel des Diffusanteils zu verändern. Hierbei wird der Diffusanteil bereits zuvor mit einem Faktor gewichtet, damit bei einer Wahl von „0dB“ der Diffuseindruck bei unbearbeiteter und bearbeiteter Version in etwa gleich bleibt. In welchem Verhältnis der Direkt- und Diffusanteil zueinander stehen, wird durch den Diffusheitsparameter Ψ bestimmt. Dazu wird der Direktanteil mit $\sqrt{1 - \Psi}$ und der Diffusanteil mit $\sqrt{\Psi}$ gewichtet.

Um wieder zurück auf die Ambisonics Kanäle zu kommen, werden die Lautsprechersignale anschließend mit der inversen Decodermatrix multipliziert.

Da es für Ambisonics kein genormtes Format existiert, ist hier zu beachten, dass die vom MatLab Programm ausgegebenen Daten dem erwarteten Format des Ambisonics Decoders entspricht. Besonders ist hier ein Auge auf die verwendete Normalisierung der Ambisonics Kanäle zu werfen. Da sich in zunehmendem Maße das ambiX¹² Format etabliert ([NZDS11]), wird in dieser Arbeit die von ambiX vorgeschlagene SN3D-Normalisierung¹³ verwendet. Da die verwendete MatLab Funktion zum Auswerten der Kugelflächenfunktionen die Werte aber in N3D Normalisierung¹⁴ liefert, müssen die höheren Ordnungen noch auf dieses Format gebracht werden. Die Beziehung zwischen N3D und SN3D Normalisierung lautet (vgl [sn3])

$$N_{l,m}^{N3D} = N_{l,m}^{SN3D} \sqrt{2l + 1} \quad (17)$$

Wobei l die derzeitige Ordnung angibt. Durch die Verwendung der SN3D Normalisie-

12. Ambisonics exchangeable

13. Schmidt semi-normalisation

14. Full 3-D Normalization

ung wird ebenfalls gewährleistet, dass bei reinen Punktquellen keine höhere Ordnung einen größeren Peak-Wert als das Signal 0. Ordnung hat (vgl [sn3]).

Schließlich wird jeder Kanal einzeln durch die Verwendung der ICQT¹⁵ zurück in den Zeitbereich transformiert. Hier werden vor der Rücktransformation die zuvor abgespeicherten und unbearbeiteten Daten über der Grenzfrequenz wieder hinzugefügt. Dann wird der aktuellen Bearbeitungs-Block in eine WAVE Datei geschrieben und am Ende alle erstellten WAVE Dateien zu einer Datei zusammengefügt.

5.3 Funktionen innerhalb des MatLab Programms

Innerhalb des MatLab Programms werden folgende Funktionen chronologisch aufgerufen:

[dec,phi_ls,theta_ls] = decoder(order,nmax)

Liefert die Decodermatrix einer idealen Lautsprecheranordnung und die Positionen der Lautsprecher. Order bestimmt hierbei die gewünschte Ambisonics Ordnung und nmax bestimmt wie viele Lautsprecher verwendet werden.

Xcq = cq(x, B, fs, fmin, fmax, varargin)

Diese Funktion führt die CQT durch und wurde von Christian Schörkhuber geschrieben. Als Argumente wird das zu transformierende Signal, die Anzahl der Bins pro Oktave, die Sampling Rate, die minimal und die maximal zu analysierende Frequenz übergeben. Es können noch zusätzlich weitere Parameter übergeben werden. Welche genau, kann direkt in der Funktionsbeschreibung des MatLab Skripts nachgelesen werden. Als Ausgabe erhält man ein Struct¹⁶, welches unter anderem die Transformationskoeffizienten enthält. Diese Funktion wird für alle Kanäle aufgerufen.

Y_sh_r = sh_matrix_real(order,phi_freq,theta_freq)

Wertet die Kugelflächenfunktionen der gewünschten Ordnung an der übergebenen Position aus, wobei hier der Winkel φ im Gegensatz zum im Abschnitt 2.2 definierten Kugelkoordinatensystem im Uhrzeigersinn angegeben wird. Daher muss bei Verwendung des unter 2.2 definierten Koordinatensystems $-\varphi$ statt φ übergeben werden.

b_sh = icqt(cqW) Die inverse CQT führt die Rücktransformation in den Zeitbereich durch. Diese Funktion erwartet als Variable ein Struct, welches gleich aufgebaut ist wie jenes, das die CQT Funktion erzeugt hat.

15. Inverse Constant-Q-Transformation

16. Structure Array - Ein Datentyp in MatLab, welcher mehrere Datentypen gruppiert

5.4 Die Bedienoberfläche des Programms

Zur besseren Bedienbarkeit des Programms soll ein User Interface implementiert werden, welches auch die Bedienung für Musiker erleichtern soll. Für das Erstellen der GUI wurde der Matlab interne GUI Designer verwendet.

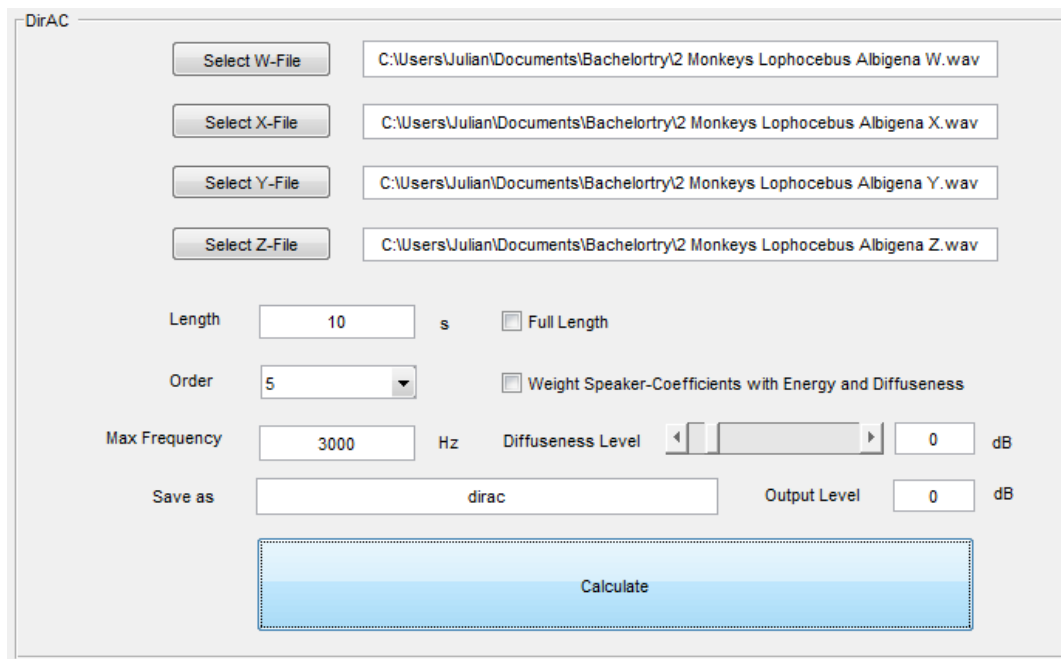


Abbildung 5: User Interface

Folgende Parameter können verändert werden:

„Length“ bestimmt wie viel von den ausgewählten Files (in Sekunden) verwendet wird. Möchte man z.B. nur ein kurzes Preview ist es empfehlenswert, diese Möglichkeit zu nutzen. Will man die volle Länge der Files bearbeiten, ist unter „Full Length“ ein Häkchen zu setzen.

„Order“ bestimmt die Ambisonics Ordnung, in der die Punktquellen für den Direktanteil resynthetisiert werden. Je höher die Ordnung, desto rechenaufwändiger wird die Bearbeitung, aber umso besser wird die Richtung geschärft. Derzeit ist eine Schärfung bis maximal 5. Ordnung implementiert (Default: 3).

Unter „Options -> Multi Channel-File“ ist es möglich das Format des Input-Files zu ändern. Ist „Multi Channel-File“ nicht aktiviert, muss für W-,X-,Y- und Z-Kanal je der Pfad zu den einzelnen Kanälen ausgewählt werden. Ist „Multi Channel-File“ aktiviert, muss nur ein Multi-Channel-File ausgewählt werden (Default: Deaktiviert). Die Files können durch das Drücken des entsprechenden Select-Buttons gewählt werden.

„Weight Speaker-Coefficients with Energy and Diffuseness“ bietet die Möglichkeit zwischen zwei unterschiedliche Methoden zur Glättung der Lautsprecher-Koeffizienten zu wählen. Ist diese Option gewählt, so werden die Koeffizienten mit der Energie und Diffusität gewichtet und anschließend normiert. Ist die Option nicht ausgewählt, werden die Koeffizienten normal geglättet (siehe Abschnitt 3.2.1).

„Max Frequency“ bestimmt die höchste Frequenz, welche durch den DirAC Algorithmus bearbeitet wird. Über dieser Frequenz wird das Signal unbearbeitet durchgelassen. Die maximal wählbare Frequenz entspricht der Hälfte der gewählten Sampling Frequenz. Empfohlen wird hier eine Frequenz um die 3000Hz (Default: 3000Hz). Je höher die gewählte Maximalfrequenz, desto höher die benötigte Rechendauer und desto wahrscheinlicher ist das Auftreten von einem akustisch perzeptiv wahrnehmbaren "Flirren".

Da der Direktanteil unabhängig vom Diffusanteil reproduziert wird, ist es möglich den Pegel des Diffusanteils mittels dem „Diffuseness Level“ Slider bzw. durch direkte Eingabe des Werts in das Textfeld anzupassen. Dabei reicht der Wertebereich des Diffusanteils von -Inf dB (keine Diffusheit) bis +20dB (Default: 0dB).

Neben „Save as“ ist es möglich den gewünschten Namen der Ausgabe-Datei zu bestimmen.

Mittels „Output Level“ kann der Pegel des Ausgabefiles angepasst werden. Durch die Bearbeitung des Programms kann es bei bereits stark ausgesteuerten Input-Dateien zum Übersteuern der Ausgabedatei kommen, wobei hier MatLab eine Warnung ausgibt. Hier ist also gegebenenfalls eine Verringerung des Ausgangspegels notwendig.

6 Binaurale Wiedergabe in Reaper

Für die Überprüfung der erhaltenen Ergebnisse sollte eigentlich ein Ambisonics System mit genügend Lautsprechern vorhanden sein. Da dies im Rahmen dieser Arbeit nicht immer möglich war bzw. auch für zukünftige Anwendungen des Programms nicht immer möglich sein wird, wird in diesem Abschnitt eine Alternative dazu vorgestellt.

Matthias Kronlachner hat ein VST Plugin entwickelt, welches eine binaurale Abhörmöglichkeit für Ambisonics Signale ermöglicht. Als DAW ¹⁷ bietet sich auf Grund der umfangreichen Routingmöglichkeiten Reaper ¹⁸ an. Ebenfalls benötigt wird die frei erhältliche Plugin-Suite von Matthias Kronlachner, Binaural Decoder Presets (downloadbar

17. Digital Audio Workstation

18. Downloadbar unter <http://www.reaper.fm/download.php>

unter <http://www.matthiaskronlachner.com/?p=2015>) und ein Kopfhörer.

Als erster Schritt müssen die Audioplugins in den Reaper Plugin Ordner installiert werden. Nach dem Starten von Reaper wird eine neue Audiospur erstellt und das vom Mat-Lab Programm ausgegebene Multikanal-File geöffnet. Dann wird das Plugin für die passende Ordnung ausgewählt und geladen. Wichtig ist hier auch, dass in den Routing-Einstellungen der Spur mindestens so viele „Track channels“ gewählt werden, wie Kanäle im Ambisonics File vorhanden sind. Anschließend ist noch eine der heruntergeladenen Binaural Decoder Presets zu wählen. Bei anschließender Wiedergabe kann das Ambisonics-Signal nun binaural abgehört werden.

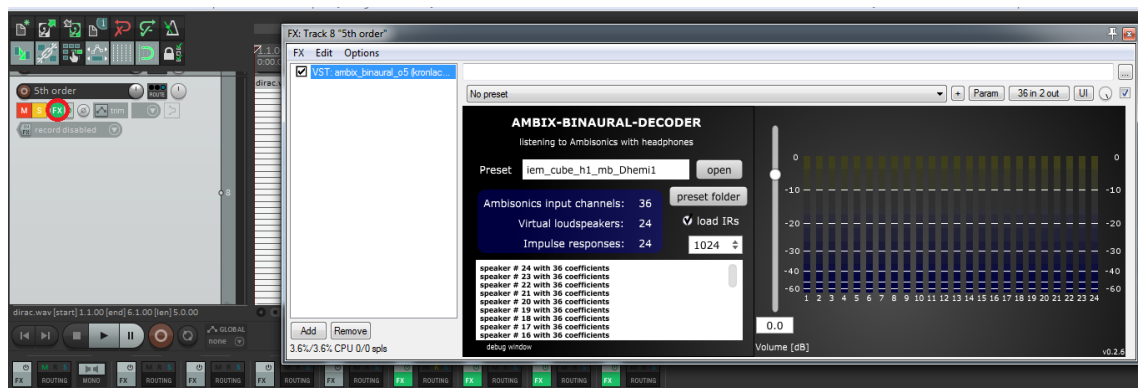


Abbildung 6: Kronlachner Plugin zur binauralen Abhöre von Ambisonics Dateien 5. Ordnung

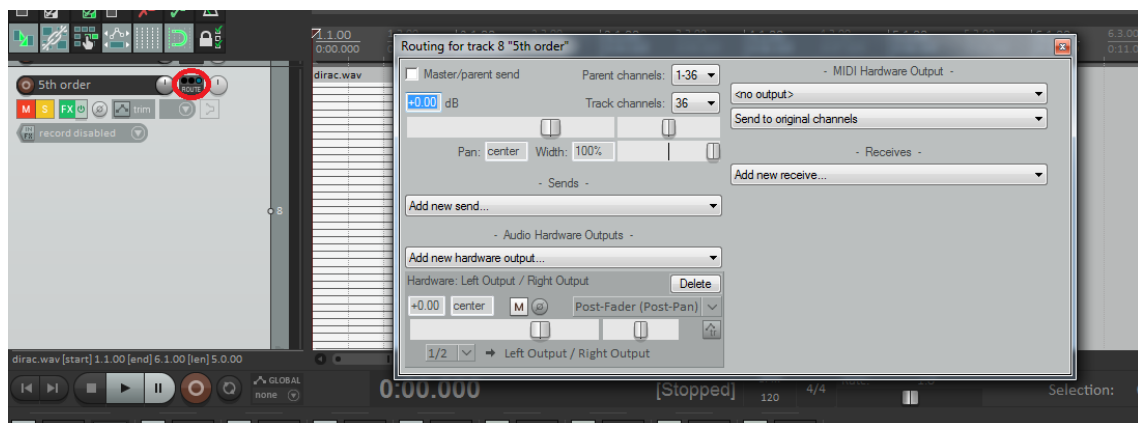


Abbildung 7: Routing Einstellungen in REAPER zur binauralen Abhöre von Ambisonics Dateien

7 Schlussfolgerung und Ausblick

Im Zuge dieser Arbeit wurde ein MatLab Programm geschrieben, welches, im Gegensatz zu dem üblichen Ansatz des Vector Based Amplitude Panning, auf Ambisonics und somit auf Kugelflächenfunktionen zurückgreift. Dabei werden mit Hilfe des Intensitätsvektors Richtungsinformationen aus dem Eingangssignal gewonnen und anschließend die Sphärisch Harmonischen an den analysierten Positionen ausgewertet. Das Signal wird vor der Analyse in Zeit- und Frequenzabschnitte eingeteilt. Dafür wurde eine gehörangepasste Analysemethode verwendet (CQT).

Dadurch, dass bei Ambisonics jedoch keine einzelnen Lautsprecher mehr angesteuert werden können, konnten die Methoden zur Glättung des Richtungsvektors und zur Erzeugung der Diffusität, welche in der Literatur ([FP06], [Pul07], [LKDP11]) vorgeschlagen wurden, nicht direkt übernommen werden. So ergaben sich Probleme mit der Glättung der Richtung und der Erzeugung des Diffusanteils. Zur Lösung dieser Probleme wurden die Ambisonics Signale auf eine virtuelle, ideale Lautsprecheranordnung projiziert, um anschließend die in der Literatur vorgeschlagenen Methoden zu verwenden. Obwohl diese Vorgehensweise zu größtenteils befriedigenden Ergebnissen führt, könnten insbesondere bei der Reproduktion der Diffusheit noch Verbesserungen durchgeführt werden.

Für die Glättung der Lautsprecherkoeffizienten wurden zwei verschiedene Ansätze implementiert, aus denen einer gewählt werden kann. Der erste Ansatz liefert ein ausgeglichenes Klangbild und gewährleistet eine passende Energiebilanz des Direktanteils, bezogen auf das ursprüngliche Signal. Dafür wird jedoch den höheren Ordnungen Energie entzogen, was zu einer nicht optimalen Richtungsschärfung führt. Der zweite Ansatz bringt ein künstlicheres Klangbild und muss regelmäßig manuell angepasst werden, schärft dafür aber die Richtung effektiver. Auch hier besteht das Potential für Verbesserungen. Zusammenfassend kann gesagt werden, dass das Programm gut und effektiv funktioniert, aber an den erwähnten Stellen noch verbessert werden kann.

Literatur

- [BC05] E. Benjamin and T. Chen, “The native b-format microphone,” in *Audio Engineering Society Convention 119*, Oct 2005. [Online]. Available: <http://www.aes.org/e-lib/browse.cfm?elib=13348>
- [FP06] C. Faller and V. Pulkki, “Directional audio coding: Filterbank and stft-based design,” in *Audio Engineering Society Convention 120*, May 2006. [Online]. Available: <http://www.aes.org/e-lib/browse.cfm?elib=13462>
- [FZS15] M. Frank, F. Zotter, and A. Sontacchi, “Producing 3d audio in ambisonics,” in *Audio Engineering Society Conference: 57th International Conference: The Future of Audio Entertainment Technology - Cinema, Television and the Internet*, Mar 2015. [Online]. Available: <http://www.aes.org/e-lib/browse.cfm?elib=17605>
- [Hos11] K. Hostniker, “Analysis and re-synthesis of directional spatial impulse responses,” Master’s thesis, KUG Graz, 2011.
- [LKDP11] M.-V. Laitinen, F. Kuech, S. Disch, and V. Pulkki, “Reproducing applause-type signals with directional audio coding,” *J. Audio Eng. Soc.*, vol. 59, no. 1/2, pp. 29–43, 2011. [Online]. Available: <http://www.aes.org/e-lib/browse.cfm?elib=15774>
- [MP05] J. Merimaa and V. Pulkki, “Spatial impulse response rendering i: Analysis and synthesis,” *J. Audio Eng. Soc.*, vol. 53, no. 12, pp. 1115–1127, 2005. [Online]. Available: <http://www.aes.org/e-lib/browse.cfm?elib=13401>
- [NZDS11] C. Nachbar, F. Zotter, E. Deleflie, and A. Sontacchi, “Ambix - a suggested ambisonics format,” in *Ambisonics Symposium*, 2011. [Online]. Available: http://iem.kug.ac.at/fileadmin/media/iem/projects/2011/ambisonics11_nachbar_zotter_sontacchi_deleflie.pdf
- [PM06] V. Pulkki and J. Merimaa, “Spatial impulse response rendering ii: Reproduction of diffuse sound and listening tests,” *J. Audio Eng. Soc.*, vol. 54, no. 1/2, pp. 3–20, 2006. [Online]. Available: <http://www.aes.org/e-lib/browse.cfm?elib=13664>
- [Pul07] V. Pulkki, “Spatial sound reproduction with directional audio coding,” *J. Audio Eng. Soc.*, vol. 55, no. 6, pp. 503–516, 2007. [Online]. Available: <http://www.aes.org/e-lib/browse.cfm?elib=14170>
- [SKHD14] C. Schörkhuber, A. Klapuri, N. Holighaus, and M. Dörfler, “A matlab toolbox for efficient perfect reconstruction time-frequency transforms with log-frequency resolution,” in *Audio Engineering Society Conference: 53rd International Conference: Semantic Audio*, Jan 2014. [Online]. Available: <http://www.aes.org/e-lib/browse.cfm?elib=17112>
- [sn3] “Ambisonics data exchange formats,” Wikipedia. [Online]. Available: https://en.wikipedia.org/wiki/Ambisonic_data_exchange_formats
- [IS11] F. Iglisch and J. Spötl, “Soundfield microphone (b-format),” Bachelor thesis, KUG Graz, 2011.