

Bachelorarbeit

im Studiengang Audiovisuelle Medien

Workflows zur binauralen Audioproduktion in Ableton Live:

Untersuchung ausgewählter Softwarelösungen

vorgelegt von

Timo Haas

38293

an der Hochschule für Medien Stuttgart

am 28.04.2025

zur Erlangung des akademischen Grades Bachelor of Engineering (B. Eng.)

Erstprüfer: Prof. Dr. Frank Melchior

Zweitprüfer: Prof. Olivier Curdt

Kurzfassung

Diese Arbeit untersucht die Integration von dynamischer Binauralsynthese in den Produktionsworkflow von Ableton Live. Ziel war es, verschiedene Softwarelösungen zu analysieren und ihre Stärken und Schwächen in der praktischen Anwendung zu vergleichen. Dafür wurden drei unterschiedliche Ansätze untersucht: Envelop For Live, XP4L und SPAT Revolution. Der Fokus lag dabei auf der Integration der Lösungen in einen Ableton Live-Workflow mit Headtracking und zentralen Funktionen wie der Bewegungssteuerung, dem Umgang mit verschiedenen Quellformaten, der Gruppierung mehrerer Elemente und der Raumsimulation. Ein Beispielprojekt wurde konzipiert, an dem die Ansätze praktisch getestet und entlang definierter Kriterien miteinander verglichen wurden.

Alle Lösungen ermöglichten grundsätzlich eine binaurale Produktion mit Headtracking in Ableton Live, unterschieden sich aber deutlich in der Einbindung, Handhabung und den Gestaltungsmöglichkeiten. Die Ergebnisse bieten eine Orientierungshilfe für Produzierende bei der Auswahl geeigneter Lösungen für die Umsetzung binauraler Workflows in Ableton Live.

Abstract

This paper examines ways to incorporate a binaural production workflow into the environment of Ableton Live. For this purpose, three tools were tested: Envelop For Live, XP4L, and SPAT Revolution. The focus was on integrating the tools into a workflow with headtracking and testing critical functions such as movement control and reverberation. The tools were evaluated by testing them within a practical project and comparing them across defined criteria.

In summary, all three tools proved to be useful inside a binaural production process using headtracking in Ableton Live. The main differences were found to be how they integrated into the existing workflow and what possibilities they offered for shaping a spatial scene.

The results provide a guide to producers wanting to implement binaural production into their Ableton workflow.

Inhaltsverzeichnis

1. Einleitung.....	1
2. Grundlagen.....	2
2.1 Grundlagen der Binauralsynthese	2
2.1.1 Räumliches Hören.....	2
2.1.2 Binauralsynthese	4
2.2 Ambisonics.....	5
2.2.1 Grundprinzipien von Ambisonics.....	5
2.2.2 First-Order Ambisonics (FOA)	5
2.2.3 Higher-Order Ambisonics (HOA).....	5
2.2.4 Enkodierung und Dekodierung	6
2.3 Dynamische Binauralsynthese und Headtracking.....	7
3. Produktionsumgebung.....	8
3.1 Ableton Live.....	8
3.2 Max MSP / Max For Live	8
3.3 Ircam SPAT~	9
3.4 Headtracking-System	10
4. Beschreibung der Ansätze	11
4.1 Envelop For Live.....	11
4.1.1 Routing und Signalfluss	11
4.1.2 Geräteübersicht und Funktionen	12
4.2 XP4L	15
4.2.1 Routing und Signalfluss	15
4.2.2 Geräteübersicht und Funktionen	16
4.3 SPAT Revolution.....	19
4.3.1 Signalfluss und Routing	19
5. Anwendungsbeispiel	23
5.1 Beispielprojekt in Ableton Live	23

5.2 Anforderungen zur Umsetzung	24
6. Anwendung im Beispielprojekt.....	25
6.1 Envelop for Live	25
6.2 XP4L	28
6.3 SPAT Revolution.....	31
7. Vergleich und Diskussion.....	34
7.1 Setup und Integration in Ableton	34
7.2 Bewegungssteuerung.....	35
7.3 Unterschiedliche Quellformate	36
7.4 Gruppierung mehrerer Elemente.....	37
7.5 Kontrolle über Raumsimulation.....	37
7.7 Headtracking Integration.....	38
8. Fazit.....	39
Abbildungsverzeichnis.....	41
Literaturverzeichnis.....	42
Ehrenwörtliche Erklärung	45

1. Einleitung

Kopfhörer sind im Alltag durch die weite Verbreitung von Smartphones und dem großen Angebot an Musik-Streaming längst zu einem gängigen Medium zum Konsum von Audioinhalten geworden. Laut Daten von Futuresource (2024) ist die Zahl der weltweit verkauften Kopfhörer von 309,5 Millionen im Jahr 2013 auf 586 Millionen im Jahr 2024 gestiegen. Gleichzeitig verändern sich auch die angebotenen Inhalte. Immersive Audioformate, die einen räumlichen Höreindruck ermöglichen, gelangen über Bereiche wie VR- und AR-Anwendungen, Gaming und Film immer mehr in den Mainstream. Auch Musikstreaming-Plattformen wie Apple Music und Tidal bieten vermehrt Inhalte in immersiven Audioformaten an, was Künstlerinnen und Künstlern vermehrt dazu motiviert, Inhalte auch in solchen Formaten zu produzieren.

Eines der Hindernisse bei der Verbreitung von 3D-Audio Inhalten ist die Verfügbarkeit der Wiedergabetechnik. Die passenden Lautsprechersysteme sind für viele interessierte Konsumierende schwer zugänglich. Binaurale Wiedergabeverfahren ermöglichen hingegen das Abspielen räumlicher Audioproduktionen über handelsübliche Kopfhörer. Da Kopfhörer als Abspielmedium schon weit verbreitet sind, ist es möglich die Inhalte darüber einer breiten Masse von Hörenden zur Verfügung zu stellen.

Für die Produktion werden deshalb Workflows zur Erstellung von Inhalten für die binauralen Wiedergabe über Kopfhörer benötigt. Ableton Live ist eine weit verbreitete Digital Audio Workstation (DAW) und besonders unter Künstlerinnen und Künstlern beliebt, die elektronische Musik produzieren. Um die Produktion von binauralen Inhalten auch in Ableton Live zu ermöglichen, wurden verschiedene Softwarelösungen entwickelt. Diese Lösungen werden in dieser Arbeit analysiert und anschließend anhand eines Beispielprojekts in Ableton Live verglichen, um ihre Stärken und Schwächen im Hinblick auf konkrete Produktionsszenarien aufzuzeigen.

2. Grundlagen

2.1 Grundlagen der Binauralsynthese

2.1.1 Räumliches Hören

Das menschliche Hören ermöglicht uns nicht nur das Wahrnehmen von Schallereignissen, sondern auch eine genaue Lokalisation von Schallquellen in einer dreidimensionalen Umgebung. In natürlichen Hörsituationen wird der Schall durch verschiedene physikalische Prozesse wie Reflexionen, Absorption und Beugung beeinflusst, bevor er die Trommelfelle erreicht. Um eine räumliche Zuordnung von Schallquellen zu ermöglichen, verarbeitet das auditive System verschiedene Merkmale der Ohrsignale, also der Signale, die schließlich an beiden Trommelfellen ankommen. Diese Ohrsignale enthalten sowohl interaurale Unterschiede als auch monaurale spektrale Hinweise, die gemeinsam eine präzise Lokalisation von Schallquellen ermöglichen (Blauert & Braasch, 2008, S. 88; Møller, 1992, S. 171).

Die beiden wichtigsten interauralen Merkmale sind die interaurale Laufzeitdifferenz (ITD) und die interaurale Pegeldifferenz (ILD). Die ITD beschreibt den zeitlichen Unterschied, mit dem ein Schallereignis beide Ohren erreicht. Sie ist für tieffrequente Signale besonders relevant, da das menschliche Gehör in diesem Frequenzbereich besonders empfindlich auf Phasenunterschiede reagiert. Für reine Sinustöne liegt die obere Grenzfrequenz, ab der interaurale Zeitdifferenzen nicht mehr zuverlässig ausgewertet werden können, bei etwa 1,6 kHz (Blauert & Braasch, 2008, S. 95–97). Die ILD beschreibt Pegelunterschiede zwischen beiden Ohrsignalen, die insbesondere durch Abschattung des Kopfes entstehen. Dieser Effekt wird mit steigender Frequenz stärker, da höhere Frequenzen aufgrund ihrer kurzen Wellenlänge weniger stark um den Kopf gebeugt werden können und deshalb auf der vom Schallereignis abgewandten Seite abgeschwächt ankommen (Blauert & Braasch, 2008).

Neben den interauralen Differenzen spielen auch monaurale spektrale Hinweise eine essenzielle Rolle für die räumliche Wahrnehmung. Diese entstehen durch die Wechselwirkung des einfallenden Schalls mit dem Kopf, dem Oberkörper und vor allem der individuellen Form der Ohrmuscheln (Pinnae). Die komplexe Geometrie der Pinnae führt dazu, dass bestimmte Frequenzbereiche je nach Einfallswinkel des Schalls verstärkt oder abgeschwächt werden. Dadurch entstehen charakteristische Muster, die vom Gehör zur Lokalisierung von Schallquellen genutzt werden (Møller, 1992, S. 171). Diese spektralen Hinweise sind besonders relevant, wenn sich die Schallquelle in der Medianebene des Kopfes befindet (siehe Abb.1). In diesen Fällen liefern ITDs und ILDs kaum verlässliche Informationen, sodass das Gehirn verstärkt auf spektrale Veränderungen angewiesen ist. Da diese Muster durch die Anatomie des Hörers geprägt sind,

unterscheiden sich die spektralen Modifikationen von Person zu Person. Die Unsicherheit durch das Fehlen der interauralen Differenzen in diesem Fall kann zumindest teilweise dadurch aufgelöst werden, dass ständig bewusste und unbewusste Peilbewegungen des Kopfes durchgeführt werden, wodurch sich die Ohrsignale auf bestimmte Weise verändern. „Wegen der Trägheit bei Kopfbewegungen kann nur dann ‚gepeilt‘ werden, wenn die Schallereignisse länger als ca. 200ms andauern.“ (Blauert & Braasch, 2008, S. 88) Die Kombination aus interauralen Differenzen und spektralen Hinweisen ermöglicht eine präzise Lokalisation von Schallquellen.

Die gesamten spektralen und zeitlichen Veränderungen, die ein Schallsignal auf seinem Weg zu den Trommelfellen erfährt, können durch sogenannte Head-Related Transfer Functions (HRTFs) beschrieben werden. Diese Übertragungsfunktionen charakterisieren die Einflüsse von Kopf, Rumpf und Ohrmuscheln und bilden damit eine zentrale Grundlage für räumliche Audiowiedergabeverfahren, insbesondere bei der Binauralsynthese (Blauert & Braasch, 2008, S. 87–90; Møller, 1992, S. 173–174).

Grundlage für diese HRTFs sind Head-Related Impulse Responses (HRIRs), die die Schallausbreitung von einer Punktquelle bis zum Gehörgang angeben. HRIRs werden üblicherweise in reflexionsarmen Räumen gemessen, wobei Miniaturmikrofone direkt in den Gehörgängen von Versuchspersonen oder eines Kunstkopfes platziert sind. Ein breitbandiges Messsignal wird aus verschiedenen definierten Positionen abgestrahlt. Aus den HRIRs lassen sich dann die Übertragungsfunktionen zwischen dem Lautsprecher und den Mikrofonen berechnen, die abhängig von Azimut, Elevation und Entfernung die Position einer Schallquelle in einem Kopfbezogenen Kugelkoordinatensystem (siehe Abbildung 1) beschreiben (Blauert & Braasch, 2008, S. 90; Brinkmann & Pike, 2020, S. 2).

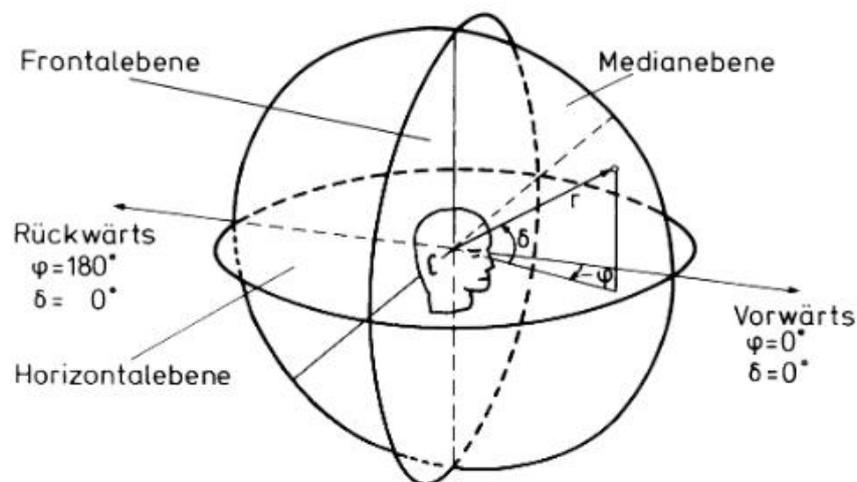


Abbildung 1: Kopfbezogenes Kugelkoordinatensystem (Blauert & Braasch, 2008, S.88)

2.1.2 Binauralsynthese

Die Binauralsynthese verfolgt das Ziel, die Mechanismen des natürlichen Hörens so nachzubilden, dass die Wiedergabe bei den Hörenden die gleiche Wahrnehmung hervorruft, als wären sie dem realen Schallfeld ausgesetzt. (Brinkmann & Pike, 2020; Møller, 1992). Dafür werden die zuvor beschriebenen Head-Related Transfer Functions (HRTFs) genutzt, um virtuelle Schallquellen räumlich zu positionieren. Das Audiosignal wird durch Faltung mit den HRTFs so bearbeitet, dass genau jene Veränderungen, die der Schall in einer realen Umgebung auf dem Weg zum Trommelfell erfahren würde, nachgebildet werden. Das Quellsignal wird also nach der Faltung in der Richtung wahrgenommen, für die die HRTF gemessen wurde. (Blauert & Braasch, 2008, S. 90)

Da die meisten Höreindrücke in Umgebungen entstehen, in denen der Schall durch Wände oder andere Objekte absorbiert und reflektiert wird, treffen neben dem Direktschall auch weitere Reflexionen beim Hörer ein (Middlebrooks, 2015). Dabei wird zwischen sogenannten Early Reflections, also frühen Reflexionen, die kurz nach dem Direktsignal eintreffen, und späteren Reflexionen, dem Nachhall, unterschieden. Vor allem die Early Reflections beeinflussen dabei wesentlich die Externalisation von Hörereignissen und Lokalisation von Quellen bei binauraler Wiedergabe. (Begault, 1992; Krebber et al., 2000) Für einen realistischen Höreindruck ist es deshalb wichtig, auch die akustische Umgebung durch Raumsimulation zu berücksichtigen.

Ein Vorläufer der Binauralsynthese ist die Kunstkopfstereofonie, bei der Mikrofone an die Gehörgänge eines künstlichen Kopfes platziert werden. Diese Methode liefert schon ein binaurales Signal, ist allerdings an die jeweilige Aufnahmeumgebung gebunden und erlaubt nachträglich kaum Modifikationen der räumlichen Eigenschaften (Møller, 1992, S. 171–172). Im Gegensatz dazu bietet die Binauralsynthese den Vorteil, virtuelle akustische Szenen flexibel und dynamisch gestalten zu können. Positionen und Bewegungen von Schallquellen sowie die Eigenschaften der akustischen Umgebung können in Echtzeit angepasst werden.

Die Wiedergabe erfolgt über Kopfhörer, da so sichergestellt werden kann, dass jedes Ohr nur das für dieses Ohr berechnete Signal empfängt. Bei der Wiedergabe über Lautsprecher würde ein unerwünschtes Übersprechen entstehen, da das Signal jedes Lautsprechers von beiden Ohren wahrgenommen wird (Møller, 1992, S. 172). Es ist jedoch theoretisch möglich binaurale Signale über Lautsprecher wiederzugeben, indem beim Abspielen des Signals eine sogenannte Crosstalk-Cancellation durchgeführt wird (Transaurale Wiedergabe) (Eargle, 1996). Dieses Verfahren wird im Rahmen dieser Arbeit jedoch nicht weiter berücksichtigt, da der Fokus auf der Kopfhörerwiedergabe liegt.

2.2 Ambisonics

2.2.1 Grundprinzipien von Ambisonics

Das Ambisonics-Prinzip wurde schon in den 1970er Jahren durch Arbeiten von Felgett (1974), Craven und Gerzon (1977) bekannt. Ambisonics baut auf koinzidenten Mikrofontechniken wie XY- und MS-Stereofonie auf, ermöglicht aber eine räumliche Repräsentation des gesamten Schallfelds, die unabhängig von der späteren Wiedergabekonfiguration dekodiert werden kann. Das Schallfeld wird mithilfe sogenannter sphärischer Harmonischer dargestellt. Die sphärischen Harmonischen sind mathematische Funktionen, die es ermöglichen, die Richtung und räumliche Verteilung des Schalls formatunabhängig zu speichern. Das ist insbesondere in Anwendungen wie Virtual Reality (VR) und Augmented Reality (AR) von Vorteil, bei denen Headtracking verwendet wird um dynamische Kopfbewegungen zu berücksichtigen. (Zotter & Frank, 2019, S. 1–6)

2.2.2 First-Order Ambisonics (FOA)

Für 2D-Ambisonics erster Ordnung wird das MS-Verfahren um ein Mikrofon mit Achtercharakteristik erweitert, welches die Front-Back Achse abbildet. Für 3D-Ambisonics erster Ordnung wird ein weiteres Mikrofon mit Achtercharakteristik ergänzt, das nach oben und unten gerichtet ist, sodass sich ein vollständiges dreidimensionales Schallfeld erfassen lässt. (Zotter & Frank, 2019, S. 1–6)

Das resultierende dreidimensionale Schallfeld in FOA besteht aus vier Kanälen:

W: Omnidirektionaler Kanal (Druckanteil)

X, Y, Z: Kanäle mit Achtercharakteristik, die entlang der kartesischen Achsen ausgerichtet sind

FOA ermöglicht zwar eine vollständige dreidimensionale Repräsentation des Schallfelds, ist aber in der räumlichen Auflösung noch begrenzt. Vor allem in Bezug auf die Lokalisationsschärfe und die Größe des nutzbaren Hörbereichs („Sweet Spot“). Um diesen Bereich zu erweitern und feinere Richtungsunterschiede abbilden zu können, wurde das Verfahren zu Higher-Order Ambisonics (HOA) weiterentwickelt. (Zotter & Frank, 2019, S. 18–20)

2.2.3 Higher-Order Ambisonics (HOA)

Higher-Order Ambisonics erweitert das Prinzip der sphärischen Harmonischen über das Hinzufügen weiterer sphärischer Harmonischer. Dadurch erhöht sich die Richtungsauflösung weiter, was eine präzisere Lokalisierung von Schallquellen und eine realistischere Darstellung räumlicher Tiefe, Weite und Diffusität ermöglicht. Die Anzahl benötigter Kanäle erhöht sich dabei aber schnell, wodurch bei Ambisonics dritter Ordnung bereits 16 Kanäle verwendet werden. (Zotter & Frank, 2019)

2.2.4 Enkodierung und Dekodierung

Ambisonics-Signale lassen sich nicht nur durch die Aufnahme mit Mikrofonarrays, sondern auch digital erzeugen. Über Ambisonics-Encoder werden virtuelle Schallquellen platziert und anschließend in ein mehrkanaliges Signal übersetzt. Dieses Signal teilt die räumlichen Anteile auf die einzelnen Kanäle auf und speichert sie in einem Format, das für die Weiterverarbeitung geeignet ist. Gängig ist hier beispielsweise das ambiX-Format, das eine feste Kanalreihenfolge (Ambisonic Channel Number: ACN) hat und die SN3D-Gewichtung (Schmidt Semi-Normalized) verwendet (Nachar et al., 2011; Zotter & Frank, 2019).

Bevor man das Ambisonics-Signal über Lautsprecher oder Kopfhörer wiedergeben kann, muss es dekodiert werden. Ziel dabei ist es, aus der Beschreibung des Schallfelds über die sphärischen Harmonischen jeweils ein konkretes Signal für jeden Lautsprecher des jeweiligen Wiedergabesystems zu gewinnen.

Die Dekodierung für ein binaurales Signal für Kopfhörerwiedergabe läuft in der Regel über ein virtuelles Lautsprecherlayout. Die Positionen der virtuellen Lautsprecher werden an einen Binaural-Decoder übergeben, der jedes Lautsprechersignal mit der passenden HRTF faltet. Bei einer Konfiguration von vier virtuellen Lautsprechern ergibt das insgesamt acht Faltungen, jeweils vier für jedes Ohr (Zotter & Frank, 2019, S. 85).

Dieser Ansatz hat den Nachteil, dass bei niedrigen Ambisonics Ordnungen Verfärbungen sowie ein Verlust von hohen Frequenzen und räumlicher Präzision eintreten können. Eine weitere Methode stellt die Magnitude Least Squares (MagLS) Methode dar. Dabei wird direkt für Kopfhörer optimiert, ohne davor den Umweg über Lautsprecher zu gehen. Das System legt den Fokus darauf, vor allem die Lautstärkeverteilungen der HRTFs möglichst genau nachzubilden. Dafür wird oberhalb einer Grenzfrequenz, üblicherweise bei 2-3 kHz, die Phase ignoriert, da ab diesem Frequenzbereich die Laufzeitunterschiede für die Wahrnehmung weniger wichtig werden (Schörkhuber et al., 2018; Zotter & Frank, 2019).

2.3 Dynamische Binauralsynthese und Headtracking

Eines der häufigsten Probleme bei klassischer, statischer Binauralsynthese ist das Auftreten von Front-Back-Confusions. In realen Hörsituationen werden diese Unsicherheiten durch Peilbewegungen des Kopfes aufgelöst (siehe Abschnitt 2.1.1). Bei der statischen Wiedergabe von binauralem Audio über Kopfhörer fehlt dieser Mechanismus. Eine virtuelle Quelle folgt der Bewegung des Kopfes, während eine reale Quelle statisch bleibt. Das verhindert die Wirkung der Peilbewegungen des Kopfes und erschwert damit die verlässliche Ortung von Quellen insbesondere auf der Medianebene (Thurlow & Runge, 1967; Wenzel, 2001).

Dieses Problem kann gelöst werden, indem man die Kopfbewegungen des Benutzers in Echtzeit erfasst und die virtuelle Szene entsprechend gegensteuert.

Schon Thurlow & Runge (1967) zeigen, dass Kopfbewegungen das Auftreten von Fehllokalisationen in der Horizontalebene und vor Allem die Häufigkeit von Front-Back-Confusions deutlich reduzieren. Auch Studien von Begault et al. (2001) und Hendrickx et al. (2017) bestätigen diesen Effekt, allerdings vor Allem für Quellen, die sich direkt im Bereich vor oder hinter dem Hörer befinden. Außerdem wird die Häufigkeit von In-Kopf-Lokalisation durch Headtracking stark verringert (Mackensen, 2004, S. 28; Mackensen et al., 2000).

Für eine glaubhafte Wiedergabe mit Headtracking ist die Latenz von entscheidender Bedeutung. Felderhoff et al. (1998) sagen, dass Latenzzeiten von unter 85 ms nicht wahrgenommen werden. Zwischen 85 und 101ms liegt der Übergang zwischen „wahrnehmbar“ und „nicht wahrnehmbar“, und ab 101 ms sind die Lokalisierungseffekte eindeutig wahrnehmbar. Andere Studien wie Wenzel (1998) zeigen, dass je nach Geschwindigkeit der Kopfbewegungen andere Grenzwerte gelten. In dem Falle waren es 92, 69 und 59 ms TSL (Total System Latency) für jeweils langsame, moderate und schnelle Kopfbewegungen.

Die Kopfbewegungen werden häufig in Freiheitsgraden gemessen (DOF, Degrees of Freedom). Bei 3 DOF wird erfasst, wie der Kopf sich um seine eigenen Achsen dreht. Die drei Bewegungen werden als Yaw (vertikale Achse), Pitch (horizontale Achse) und Roll (Kippen) bezeichnet. (Asperti et al., 2024) In 6 DOF kommt zusätzlich noch die Bewegung des Kopfes im Raum entlang der X-, Y- und Z-Achse.

3. Produktionsumgebung

3.1 Ableton Live

Ableton Live ist eine Digital Audio Workstation (DAW), die sich konzeptionell von klassischen DAWs unterscheidet. Mit dem Ziel eine flexible Umgebung für Live-Performance und intuitive Musikproduktion zu schaffen ist im Jahr 2000 die erste Version von Ableton Live auf den Markt gekommen. Ableton wurde über die Jahre kontinuierlich weiterentwickelt und hat sich heute als Industriestandard für Live-Performances etabliert (Margulies, 2010).

Ableton Live ist als DAW nicht primär dafür ausgelegt mit Mehrkanalaudio zu arbeiten. Die Anzahl der Kanäle pro Spur ist in Ableton standardmäßig auf Stereo festgelegt, was das zurückgreifen auf klassische Mehrkanalworkflows erschwert. Diese Einschränkung lässt sich jedoch durch die Nutzung von Max for Live umgehen.

3.2 Max MSP / Max For Live

Max/MSP

Max, auch bekannt als Max/MSP/Jitter, ist eine visuelle Programmiersprache, die vor allem in der digitalen Signalverarbeitung und in interaktiven Medienanwendungen verwendet wird. Max/MSP ist modular aufgebaut. Es werden sogenannte Patches erstellt, in denen grafische Objekte über Verbindungen miteinander kommunizieren. Diese Objekte sind die grundlegenden Bausteine der Umgebung. Die Erweiterung MSP umfasst Objekte zur Audio- und Signalverarbeitung, während Jitter für Video- und Grafikverarbeitung eingesetzt wird. (*Max 5 Help and Documentation*, o. J.) Durch sogenannte Externals, also extern programmierten Max-Objekten, sowie wiederverwendbaren Abstraktionen, Libraries und Packages von Cycling 74⁴ und der Max-Community, lassen sich komplexe Funktionen für eine große Vielfalt an Anwendungsfällen zusammenstellen. (*Externals | Cycling '74 Documentation*, o. J.)

Max for Life (M4L) ist eine in Zusammenarbeit mit Cycling 74⁴ entwickelte Erweiterung für Ableton Live, die die Integration von Max Patches innerhalb von Ableton Live ermöglicht. Damit können Nutzer beispielsweise eigene Audio- und MIDI-Effekte, Instrumente und komplexe Signalverarbeitung innerhalb der DAW einsetzen und die Funktionen von Live flexibel erweitern. Außerdem stellt M4L spezielle Objekte zur Verfügung, die sich in Optik und Verhalten direkt in die Ableton Live Benutzeroberfläche einfügen. Diese Objekte lassen sich genau wie die nativen Ableton-Parameter automatisieren, modulieren und zuweisen.

3.3 Ircam SPAT~

Der vom Forschungsinstitut IRCAM entwickelte *Spatialisateur*, oft spat~ genannt, ist eine Echtzeit-Audioengine für die räumliche Positionierung von Klangquellen (Spatialization) und zur Erzeugung von künstlichem Nachhall (Reverberation). Spat~ ist als Sammlung verschiedener Module aufgebaut, die in Max/MSP kombiniert und konfiguriert werden können. Eines der zentralen Komponenten ist die mehrkanalige Reverberation-Engine, die aus mehreren Modulen besteht, die jeweils Early Reflections, Cluster (dichte, spätere Reflektionen) und diffuse Nachhallanteile erzeugen (Carpentier et al., 2015)

Für das Panning nutzt spat~ das vielseitige Modul spat.pan~. Dieses External unterstützt eine breite Auswahl an gängigen Spatialization-Algorithmen, darunter traditionelle Stereotechniken, Binauralsynthese sowie amplitudenbasierte Verfahren wie zum Beispiel Vector-Based-Amplitude-Panning (VBAP).

Zusätzlich wird die Encodierung und Decodierung von Ambisonics, einschließlich höherer Ordnungen in 2D und 3D unterstützt. Die Decodierung kann sowohl für Lautsprecher- als auch Kopfhörerwiedergabe erfolgen. Für Binauralsynthese besteht des Weiteren die Möglichkeit der Nahfeld-Binauralsynthese, bei der HRTFs speziell für Schallquellen in unmittelbarer Nähe des Hörers angepasst werden. In zwei der getesteten Ansätze bildet spat~ die Grundlage der Signalverarbeitung.

Die Benutzeroberfläche spat.oper ermöglicht eine Steuerung der räumlichen und klanglichen Eigenschaften des virtuellen Raums sowie der Quellen. Die Bedienung erfolgt über sogenannte „Perceptual Factors“. Diese Parameter sind Bedienelemente, die jeweils mehrere technische Parameter gleichzeitig steuern. Abbildung 2 zeigt die Benutzeroberfläche von soat.oper und die Perceptual Factors einer Quelle.(Carpentier et al., 2015)

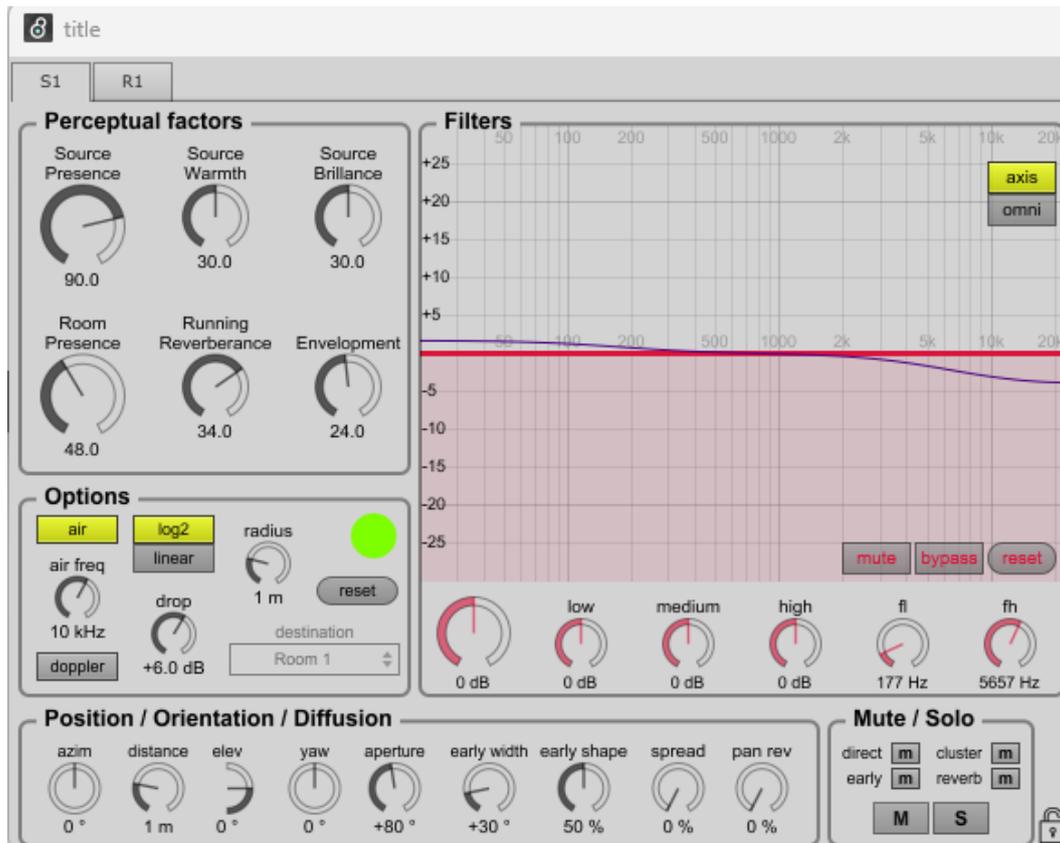


Abbildung 2: Die Benutzeroberfläche von spat.oper mit den Perceptual Factors

3.4 Headtracking-System

Für das Beispielprojekt wurde ein Headtracking-System auf Basis eines Arduino Pro Micro in Kombination mit einem BN0-55 Sensor verwendet. Die Sensordaten werden seriell (115200 Baud) über USB an die Software Headtracker OSC-Bridge von NVSonic (2019) übertragen, die daraus kontinuierlich die Kopfposition berechnet. Die Ausgabe der umgerechneten Bewegungsdaten werden über OSC weitergegeben, wobei sowohl der Port als auch die Zusammensetzung der OSC-Message in der Benutzeroberfläche angepasst werden können.

4. Beschreibung der Ansätze

4.1 Envelop For Live

Envelop For Live (*Envelop for Live*, 2018) ist ein kostenlos verfügbares Open-Source-Projekt, das aus einer Sammlung von M4L-Geräten für die Produktion mit Ambisonics dritter Ordnung in Ableton Live besteht.

Grundlage für die Ambisonics Enkodierung und Dekodierung stellen die vom ICST entwickelten *Ambisonics Externals for MaxMSP* (ICST, o. J.) dar. Dabei handelt es sich um eine Sammlung von Max/MSP Externals für die Ambisonics Verarbeitung, darunter die in E4L verwendeten *ambiencode~* und *ambidecode~*. Dadurch, dass es sich bei Envelop For Live um ein Open-Source-Projekt handelt, kann man den Source-Code und die Max Patches beliebig erweitern und für seine Anwendungsfälle anpassen. So ist es beispielsweise möglich, externe VSTs als Decoder einzubinden oder den Output als Ambisonics-Stream in externen Anwendungen weiterzuverarbeiten.

4.1.1 Routing und Signalfluss

Abbildung 3 zeigt den typischen Signalfluss von Envelop For Live. Das Stereo-Eingangssignal einer Ableton Audio- oder MIDI-Spur wird zunächst in einem der Encoder/Panner-Devices von E4L in ein Ambisonics-Signal dritter Ordnung enkodiert. Das ursprüngliche Stereo-Eingangssignal wird dabei von jedem der Geräte parallel und unbehandelt weitergegeben. Außerdem bietet jedes Gerät die Möglichkeit, zwischen zwei Optionen für das Weiterleiten des Ambisonics-kodierten Signals umzuschalten. Bei der ersten Option (E4L Master) werden die 16 Kanäle des encodierten Ambisonics-Signals direkt an den E4L Master Bus gesendet.

Die zweite Option (Chain) lässt das Gerät den Ambisonics-Stream an das nachfolgende E4L Gerät in der Kette weitergeben, um weitere Verarbeitung in der Ambisonics-Domäne zu ermöglichen (z.B. für Metering, Reverb). Erst das letzte Gerät in der Verarbeitungskette sendet das Ambisonics-Signal an den E4L Master Bus. Dort wird es entweder zur binauralen Wiedergabe für Kopfhörer oder für ein mehrkanaliges Lautsprechersystem dekodiert.

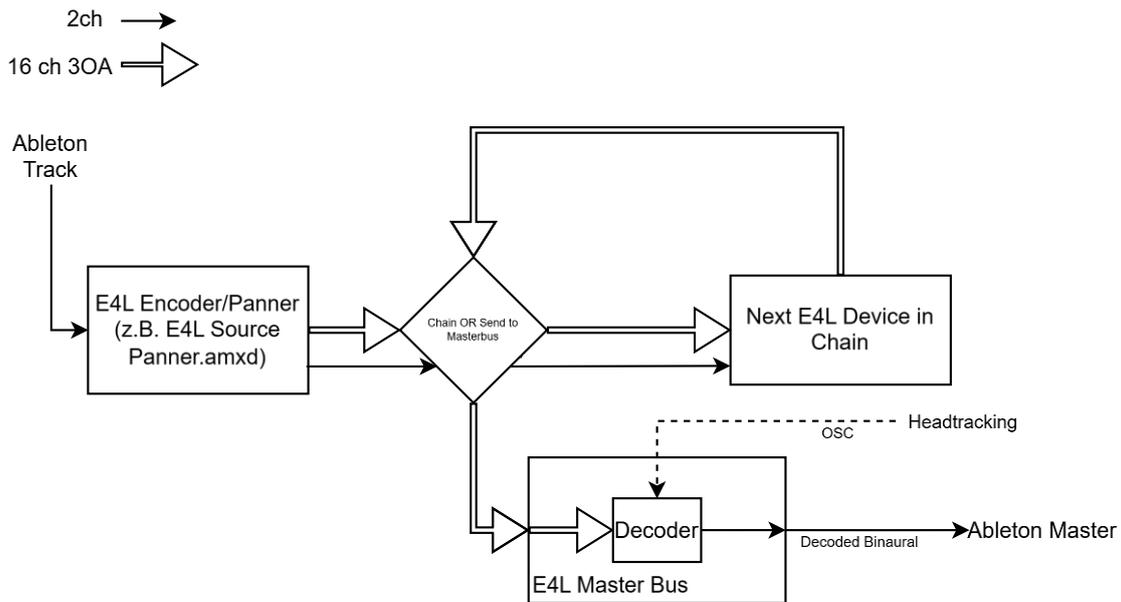


Abbildung 3: Signalfluss für binaurale Wiedergabe in Envelop For Live (eigene Darstellung)

4.1.2 Geräteübersicht und Funktionen

Der *E4L Source Panner* ist eins der Devices zur räumlichen Positionierung und Ambisonics-Enkodierung von Mono- und Stereoquellen in der E4L-Suite. Abbildung 4 zeigt die Benutzeroberfläche des *E4L Source Panners*. Über den Mono/Stereo-Schalter lässt sich zuerst festlegen, ob die beiden Eingangskanäle summiert werden oder getrennt verarbeitet werden sollen. Im Stereo-Modus kann über den Spread Parameter zusätzlich der Abstand der beiden Kanäle zueinander angepasst werden. Die Position lässt sich über die Parameter Azimut, Elevation und Radius ändern. Radius beeinflusst dabei die Lautstärke des Signals in Abhängigkeit des „Range“ Parameters. (Slee, 2024)

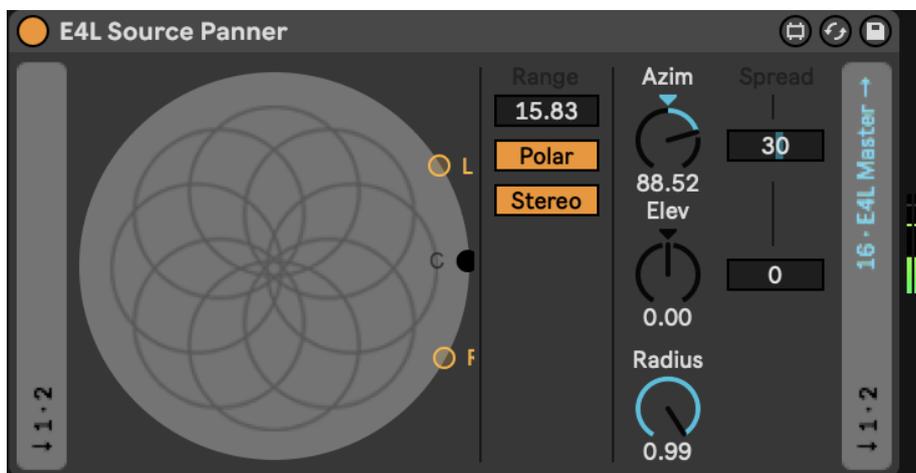


Abbildung 4: Benutzeroberfläche des *E4L Source Panners*

Das encodierte 16-Kanal-Ambisonics-Signal wird entweder direkt zum *E4L Masterbus* weitergeleitet oder an das nächste E4L Device in der Kette durchgegeben. Abbildung 5 veranschaulicht, wie das Signal eines Panners über den Chain-Modus an das E4L Meter geleitet wird, welches dann als letztes Gerät in der Kette das Signal an den Master Bus routet.



Abbildung 5: Der E4L Source Panner im "Chain" Modus

Alle von den an den E4L-Geräten gesendeten Ambisonics-Signale werden auf dem Masterbus summiert. Dieser ist außerdem zuständig für das Decoding des Ambisonics-Streams für das gewählte Ausgabeformat. Standardmäßig sind hier verschiedene Optionen für binaurales Monitoring und für Lautsprechersysteme verfügbar. Zusätzlich ist es möglich, über das AmbiX-Preset das Ambisonics Signal auszugeben und außerhalb von E4L weiterzuverarbeiten. benutzerdefinierte Decoder können hier ebenfalls als Max-Patch geladen werden.

Als Exportformat lässt sich zwischen AmbiX und FuMa sowie zwischen erster und dritter Ordnung wählen (siehe Abbildung 6).

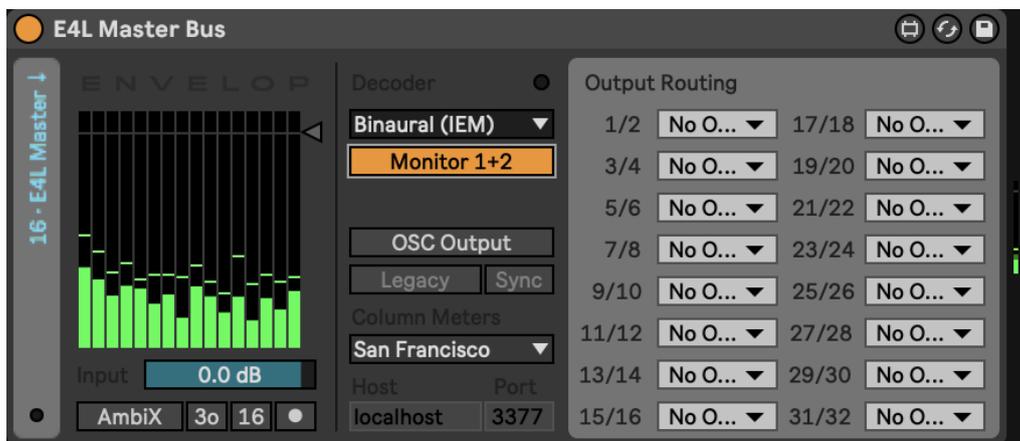


Abbildung 6: Benutzeroberfläche des E4L Master Bus

Für die Einbindung von Headtracking wurde in dieser Arbeit eine modifizierte Version des *E4L Master Bus* verwendet, die Headtracking in 3DOF über OSC integriert (siehe Abbildung 7). Dazu stehen Presets für gängige Headtracking-Systeme wie WavesNX und Supperware zur Verfügung. Die Headtracking-Daten werden zur Rotation der Ambisonics-Szene verwendet. (d5x610, 2023)

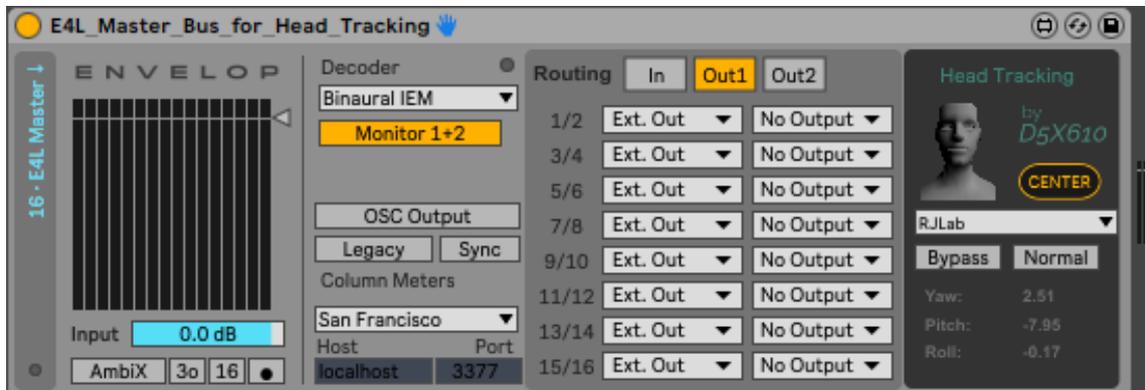


Abbildung 7: Der Modifizierte E4L Master Bus mit Headtracking Integration

Die Geräte *E4L Aux Send* und *E4L Aux Return* ermöglichen das Routen eines Ambisonics-Signal der E4L-Kette auf bis zu acht interne Aux-Busse. Dadurch lassen sich beliebig viele Spuren auf eine separate Audiospur umleiten.

Zusätzlich lassen sich über die Geräte *E4L 30A Plugin Send* und *E4L 30A Plugin Return* Drittanbieterplugins, die Ambisonics-Signale dritter Ordnung verarbeiten können, in die Kette einbinden. Dazu wird über ein *E4L 30A Plugin-Send*-Gerät das Signal an das externe Plugin übergeben und anschließend über ein *E4L 30A Plugin-Return*-Gerät zurück in den Signalfluss von E4L gebracht. Abbildung 8 zeigt eine Effektkette, bei der mehrere Signale von einem *E4L Aux Return*-Gerät empfangen, über ein eingebundenes Drittanbieterplugin verarbeitet werden und schließlich an den *E4L Master Bus* übergeben werden.

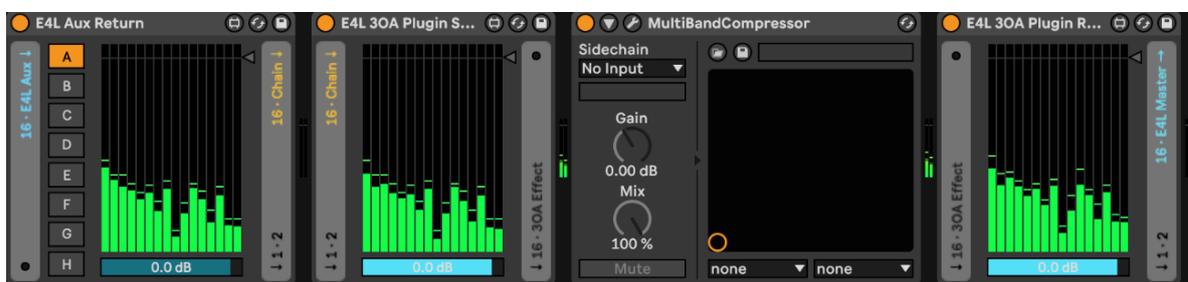


Abbildung 8: Eine Effektkette bestehend aus *E4L Aux Return*, *E4L 30A Plugin Send/Return* und *IEM MultiBandCompressor*

Das *E4L B-Format Convolution Reverb* erlaubt das Laden von Impulsantworten für jeden der FOA-Kanäle W, X, Y und Z. Die Impulsantworten können außerdem skaliert und rotiert werden. Über die Distance-Lock Funktion kann das Gerät die Informationen eines vorgeschalteten Panner-Geräts nutzen, um die Lautstärkeanteile der Early und Late Reflections an die Distanz anzupassen. Abbildung 9 zeigt die Benutzeroberfläche des *B-Format Convolution Reverbs*.

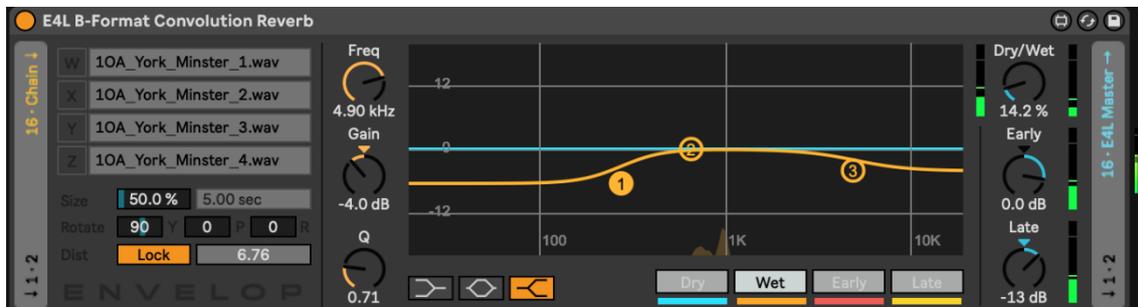


Abbildung 9: Benutzeroberfläche des *E4L B-Format Convolution Reverbs*

4.2 XP4L

XP4L (*XP for Ableton Live*, 2021) ist eine Softwarelösung, die es ermöglicht das *spat~* Framework in Ableton Live zu verwenden. Sie besteht aus mehreren M4L-Geräten, die mit einer externen Anwendung für die Visualisierung und interne Signalverarbeitung kommunizieren. Die Steuerung und das Routing finden in Ableton statt, während die Verarbeitung in der XP-Anwendung durchgeführt wird. Die Verarbeitung basiert auf Ircam *Spat~*, wodurch unter anderem die Reverb-Engine und die Panning-Module von *spat~* zur Verfügung stehen.

4.2.1 Routing und Signalfluss

XP4L benutzt verwendet mehrere M4L-Geräte und eine externe Anwendung (XP) zur Verarbeitung und Visualisierung räumlicher Audioszenen. Die Signalverarbeitung erfolgt dabei außerhalb von Ableton Live in XP. Routing, Steuerung und Parameterautomation werden dabei über die M4L-Geräte gesteuert. Der Signalfluss für den Anwendungsfall mit binauraler Wiedergabe ist in Abbildung 10 dargestellt.

Das Stereosignal einer Ableton-Audiospur wird zuerst von einem *xp.source*-Gerät aufgenommen. Dabei wird nur einer der beiden Kanäle oder eine summierte Version als Monosignal weiterverarbeitet. Das entstehende Monosignal wird einem virtuellen Objekt zugewiesen und an den Multichannel-Bus von *xp.engine* übergeben. Parallel sendet *xp.source* die Parameter für die räumliche Verarbeitung (Source Parameter) über OSC-Verbindungen an *XP*. Dazu gehören unter anderem Positionsdaten, Perceptual Factors sowie Gruppen- und Raumzuweisungen.

xp.engine dient als Multichannel-Bus für alle Eingangskanäle der bis zu 32 Source-Objekte. Dieser Bus wird an *XP* weitergegeben, wo die eigentliche Signalverarbeitung auf Basis von *spat~* erfolgt.

Im hier beschriebenen Fall eines binauralen Workflows rendert *XP* ein zweikanaliges binaurales Signal basierend auf den Steuerdaten der Geräte und leitet es an *xp.engine* zurück, wo es auf den Ableton-Mastertrack weitergegeben wird.

Zusätzlich zu den Source-Parametern von *xp.source* können über *xp.room* die Parameter der verwendeten Räume gesteuert werden. Das *xp.visual*-Gerät überträgt Headtracking-Daten mit bis zu 6 DOF, die von *XP* verarbeitet werden.

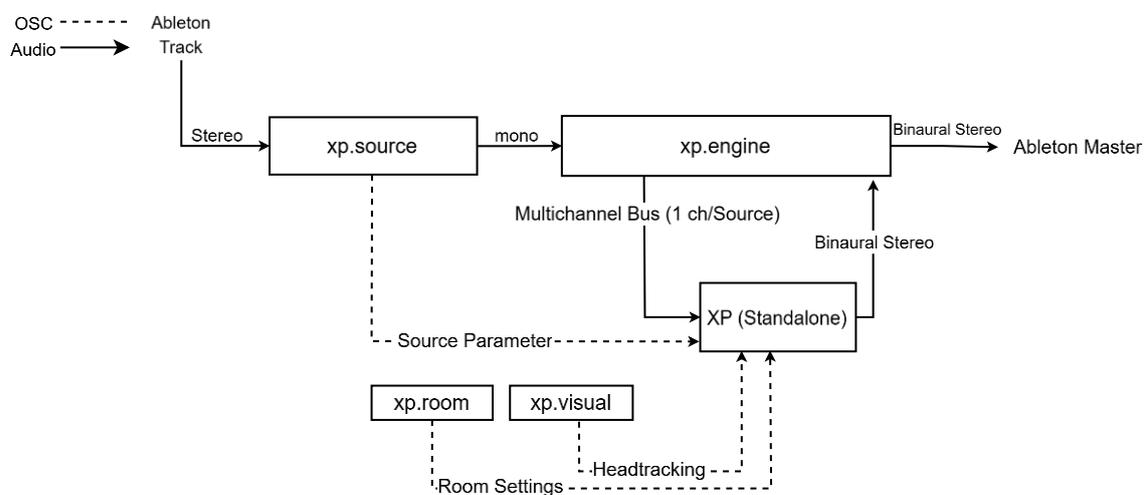


Abbildung 10: Signalfluss in XP4L bei binauraler Wiedergabe (eigene Darstellung)

4.2.2 Geräteübersicht und Funktionen

xp.source ist für das Erstellen der virtuellen Objekte und deren Eigenschaften verantwortlich. Wird *xp.source* auf eine Audio- oder MIDI-Spur geladen, wird die Spur umbenannt und in der *XP*-Anwendung zur Visualisierung erscheint ein neues Objekt mit entsprechendem Namen (z.B. source1). Dieses virtuelle Objekt ist direkt mit der Spur verknüpft. In der Benutzeroberfläche können Parameter für Position (polar oder kartesisch), Abstrahlverhalten, „Perceptual Factors“ (siehe Abschnitt 3.3), Gruppenzugehörigkeit und Raumzuweisung eingestellt werden. Abbildung 11 zeigt die Standardeinstellung von *xp.source*. Neben *xp.source* gibt es auch *xp.source.simply*, eine vereinfachte Version mit 2D-Visualisierung der Objektposition. Die Visualisierung kann auch durch ein darauffolgendes *xp.pan*-Geräts hinter ein *xp.source*-Gerät geladen werden.



Abbildung 11: Benutzeroberfläche von *xp.source* in Standardeinstellung

Für die Steuerung der Raumsimulation gibt es das Gerät *xp.room*. Es bietet Kontrolle über die Perceptual Factors der *spat~* Engine. Durch das Laden mehrerer Instanzen lassen sich bis zu vier verschiedene Räume erstellen.

Das *xp.engine*-Device wird auf eine leere Audiospur in Ableton geladen und erweitert sie zu einem Mehrkanal-Bus. Die Audiosignale aller *xp.source*-Objekte werden dort entsprechend ihrer Source-ID einem Kanal im Bus zugewiesen (z.B. source3; Input3).

In der Benutzeroberfläche werden grundlegende Einstellungen zur Projektkonfiguration vorgenommen. Der Mehrkanal-Bus mit den Audiosignalen der Quellen wird ebenfalls an XP weitergeleitet, wo die Verarbeitung stattfindet. Auch wenn der binaurale Ausgabemodus aktiviert ist, muss aufgrund der Architektur des Programms mit Fokus auf Lautsprecherwiedergabe trotzdem ein „Dummy-Speaker-Layout“ definiert werden. Abbildung 12 zeigt die Benutzeroberfläche zur Lautsprecherkonfiguration (Voreinstellung oder benutzerdefiniert), dem Spatialization Type und dem Ausgabemodus (Mute, Multichannel-Stream, Binaural).

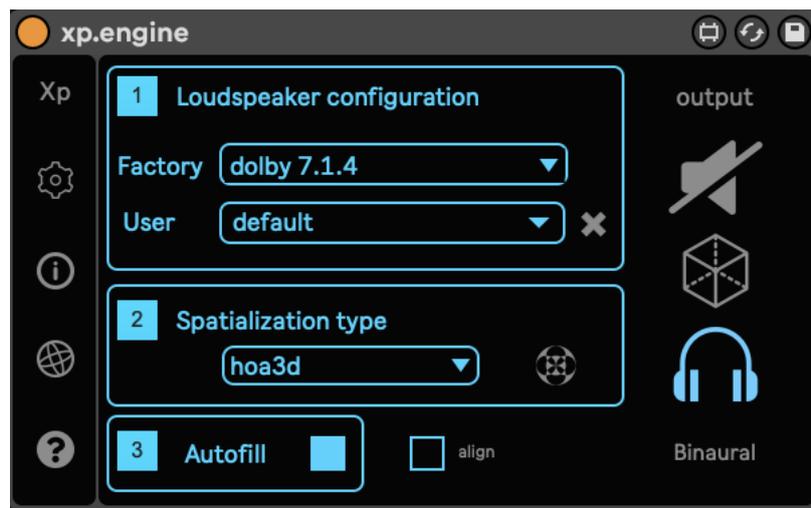


Abbildung 12: Benutzeroberfläche von *xp.engine*

Das *xp.visual*-Device ist notwendig, da es die XP-Anwendung startet. Wird es in Ableton Live auf eine Audiospur geladen, öffnet sich die Anwendung automatisch.

xp.visual dient als Fernsteuerung für die XP-Anwendung zum Wählen verschiedener Kameraansichten und Ändern des visuellen Erscheinungsbilds der Szene. Zusätzlich lässt sich der Listener-Mode aktivieren, der eine First-Person-Ansicht bietet und das Übertragen von 6DOF Headtracking-Daten über OSC erlaubt. Über die Benutzeroberfläche lässt sich Headtracking aktivieren und die Listener-Position sowie Rotation manuell ändern. Abbildung 13 zeigt die Benutzeroberfläche von *xp.visual* im Listener-Mode mit aktiviertem Headtracking.

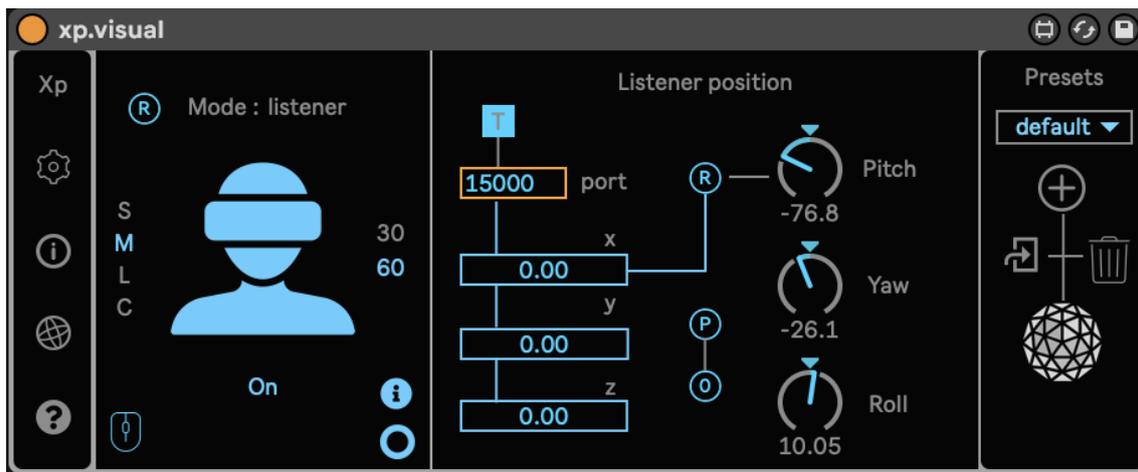


Abbildung 13: Benutzeroberfläche von *xp.visual* mit aktiviertem Headtracking

In der XP-Standalone-Anwendung laufen die Audiosignale der Quellen sowie alle OSC-Steuerdaten zusammen. Die Verarbeitung erfolgt in Echtzeit basierend auf *spat~* und hängt von den globalen Einstellungen in *xp.engine* sowie den individuellen Parametern der einzelnen Quellen, Gruppen und Räume ab. Die Anwendung wird beim Laden von *xp.visual* gestartet, und zeigt eine Echtzeit-Visualisierung der Szene wie sie in Abbildung 14 zu sehen ist.

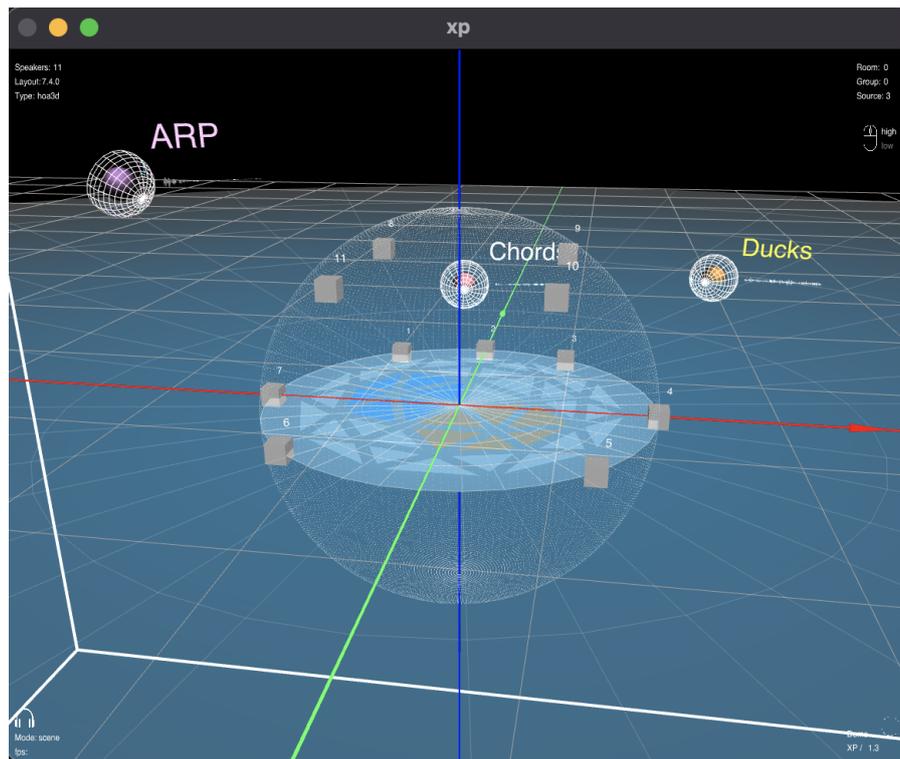


Abbildung 14: XP-Anwendung im Scene-Modus. Sichtbar sind drei Source-Objekte, positioniert durch `xp.source`

4.3 SPAT Revolution

Im Gegensatz zu E4L und XP4L ist SPAT Revolution kein auf M4L basierendes System, das speziell für die Anwendung in Ableton Live entwickelt wurde, sondern eine Anwendung, die in unterschiedlichen Produktionsumgebungen eingesetzt werden kann. Die Einbindung erfolgt je nach Setup über verschiedene Schnittstellen. In dieser Arbeit wurde SPAT Revolution über den virtuellen Audiotreiber BlackHole (*BlackHole*, 2019) sowie mithilfe der für einen Ableton Live-Workflow vom Hersteller bereitgestellten M4L-Geräte und VST-Plug-ins eingebunden.

4.3.1 Signalfluss und Routing

Abbildung 15 zeigt einen einfachen Aufbau in der Setup-Ansicht von SPAT.

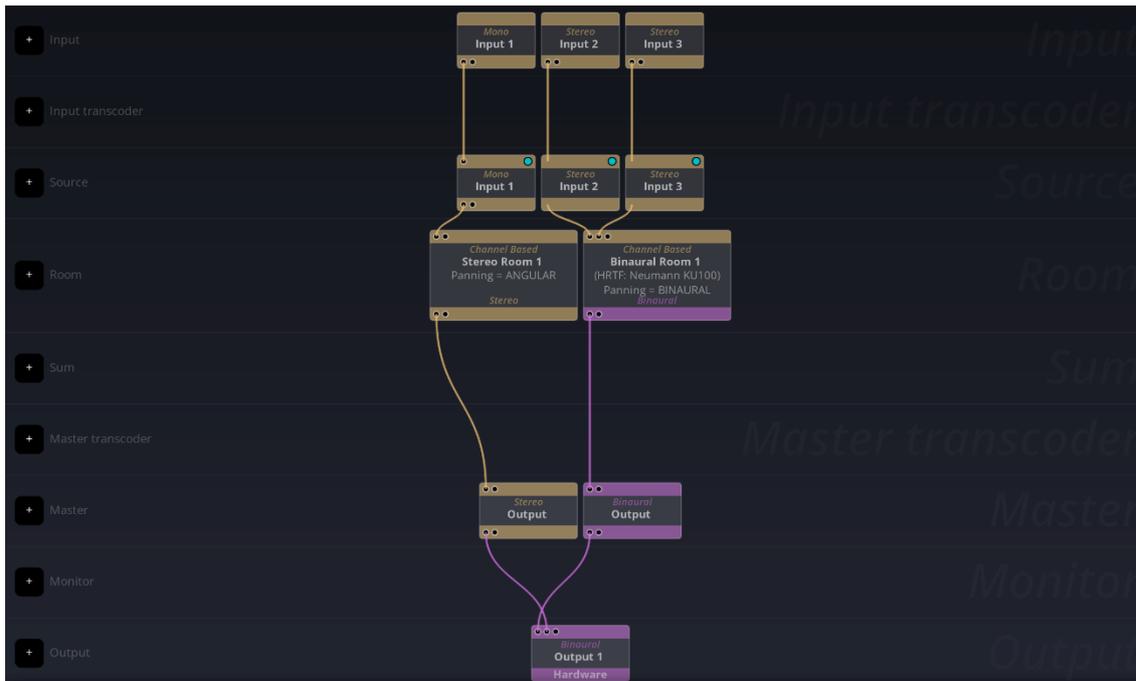


Abbildung 15: Setup-Ansicht in SPAT Revolution

Der Signalfluss in SPAT ist modular aufgebaut und wird grafisch durch hintereinandergeschaltete Modulblöcke von oben nach unten in der Setup-Ansicht dargestellt.

Er startet mit den Eingängen (Inputs). Die Eingangssignale werden von *Input-Modulen* abgegriffen. In der Konfiguration des Moduls kann der „Stream-Type“ (z.B. Channel-Based, HOA) und die entsprechenden Eingangskanäle angegeben werden. Die *Input-Transcoder-Module* ermöglichen es, zwischen unterschiedlichen Stream-Types umzuwandeln.

Die *Source-Module* in SPAT stellen die Schnittstelle zwischen den eingehenden Audiosignalen und der objektbasierten Weiterverarbeitung dar. Sie verknüpfen die Audio-Streams mit Metadaten. Diese enthalten unter anderem Informationen über die Position im Raum, Perceptual Factors und weitere Parameter der *spat*-Engine.

In den *Room-Modulen* findet die Verarbeitung statt. Sie sind vor allem durch das Ausgabeformat (Output-Stream-Type; z.B. Binaural oder Channel-Based) und die verwendeten Einstellungen für die Raumsimulation charakterisiert und erzeugen darauf basierend das räumliche Wiedergabesignal. In der Ultimate-Version von SPAT ist die Anzahl von *Room-Modulen* unbegrenzt, was parallele Renderings für unterschiedliche Formate und Monitoring-Szenarien erlaubt.

Die *Sum-Module* erlauben das Summieren verschiedener Signale des gleichen Stream-Types und die *Master-Transcoder-Module* erlauben, ähnlich wie *Input-Transcoder-Module*, das Umwandeln von verschiedenen Formaten innerhalb der Session.

Die *Master-Module* ermöglichen das Zusammenführen mehrerer Signalwege und das Routen auf verschiedene Ausgänge. Mit den *Monitor-Modulen* lassen sich verschiedene Lautsprecherkonfigurationen binaural abhören. Dabei wird die Lautsprecheranordnung virtualisiert, was sich zur Abhörkontrolle kanalbasierter Mischungen eignet.

Die *Output-Module* ermöglichen das Routing auf Ausgänge des eingestellten Ausgabe-Geräts. Auch hier können unbegrenzt viele parallele *Output-Module* verwendet werden, was die Ausgabe von verschiedenen Formaten gleichzeitig zur Wiedergabe oder zur Aufnahme erlaubt.

Um schnellen Zugang auf das Ändern der SPAT-internen Parameter innerhalb der DAW zu erhalten, bietet die Software mehrere VST-Plug-Ins, die über OSC-Verbindungen mit SPAT kommunizieren. Diese bestehen aus einem Send-, einem Return- und einem Room-Plugin. Die Plug-ins werden jeweils auf eine Audiospur geladen und enthalten eine Auswahl automatisierbarer Parameter (siehe Abbildung 16).

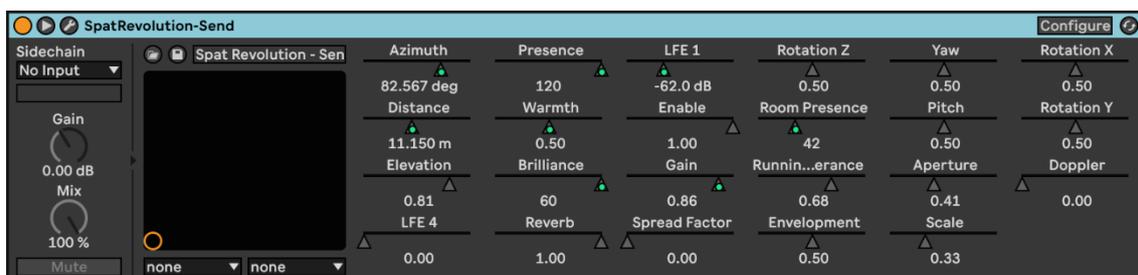


Abbildung 16: Die automatisierbaren Parameter des SpatRevolution-Send Plugins

Dafür muss in den SPAT-Voreinstellungen die Option „OSC Enable“ aktiviert sein. Um die einzelnen Verbindungen zu aktivieren, gibt es im Bereich „OSC connections“ verschiedene OSC-Connection-Presets. Ist die OSC-Verbindung aufgebaut, kann das Send-Plugin zur Automation und Steuerung der jeweiligen Source verwendet werden. Das Room-Plugin ermöglicht das Zugreifen auf Eigenschaften der jeweiligen Räume, wobei pro angelegtem *Room-Modul* ein Room-Plugin in Ableton verwendet werden kann.

Abbildung 17 zeigt den zeigt exemplarisch den Signalfluss vom Mono-Ausgangssignal eines Ableton Audio-Tracks über BlackHole durch SPAT bis hin zur binauralen Ausgabe.

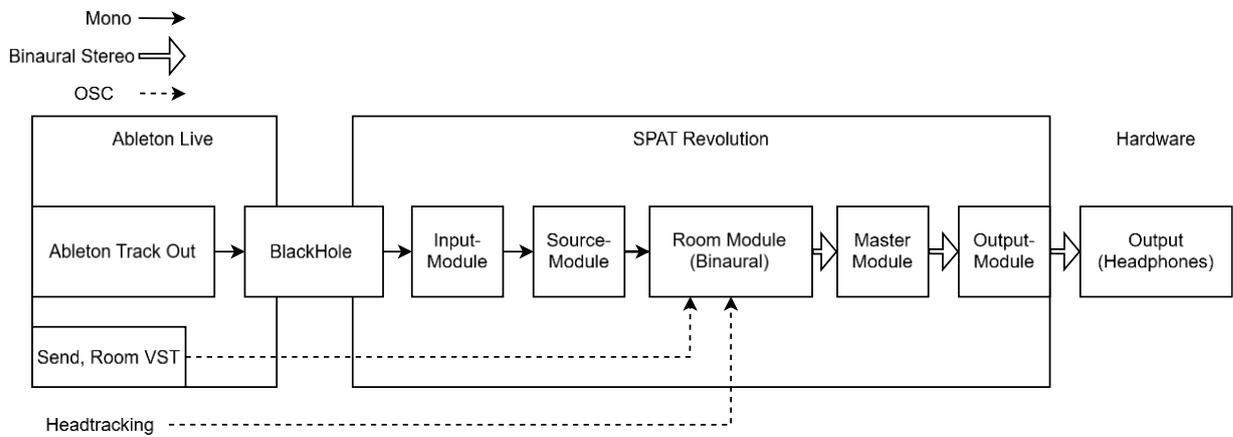


Abbildung 17: Der Signalfluss am Beispiel einer Mono-Quelle bei binauraler Wiedergabe in SPAT Revolution

5. Anwendungsbeispiel

5.1 Beispielprojekt in Ableton Live

Um die verwendeten Ansätze praxisnah zu testen, wurde ein Beispielprojekt erstellt. Dabei handelte es sich um eine Soundscape mit Elementen aus der elektronischen Musikproduktion. Das Ziel dabei war es zu veranschaulichen, welchen Einfluss die jeweiligen Setups auf Workflow und Umsetzung des geplanten Projekts haben. Dafür wurde ein Konzept aus verschiedenen Grundelementen entwickelt, die unterschiedliche Rollen im Soundscape übernehmen sollten. Die genaue Umsetzung dieser Rollen in den jeweiligen Ansätzen wurde flexibel gehalten, um die unterschiedlichen Gestaltungsmöglichkeiten und eventuell notwendigen Workarounds zu zeigen. Die Integration von Headtracking war ebenfalls vorgesehen.

Die Grundlage des Soundscapes sollten mehrere Field-Recordings bilden. Als Ausgangsmaterial waren sowohl 5.1-Surround Aufnahmen als auch First-Order-Ambisonics Aufnahmen im B-Format verfügbar.

Die Hauptelemente des Beispielprojekts waren zum einen sich durch mehrere automatisierte Filter verändernde Chords, die vor allem durch die Bewegung im Raum um den Hörer und durch räumlich verteilte Delay-Effekte charakterisiert sein sollten.

Zum anderen sollte ein Arpeggio komplexere Bewegungen um den Hörer durchführen und sich dabei abwechselnd weit entfernen und wieder annähern. Dabei sollte zwischen ruckartigen Sprüngen in der Position, schnellen Fly-Overs und langsamen Bewegungen variiert werden.

Als Bindeglied zwischen den Field-Recordings und den Chords waren zwei Layer eines Drone-Sounds in unterschiedlichen Tonhöhen geplant. Dabei handelte es sich um lang gehaltene Akkorde mit großer Stereobreite, die kontinuierlich über das Arrangement hinweg langsam um den Hörer rotieren sollten.

5.2 Anforderungen zur Umsetzung

Aus dem Beispielprojekt ergeben sich eine Reihe von Anforderungen an die verschiedenen Ansätze, um die Umsetzung der beschriebenen Rollen der Elemente zu ermöglichen.

Die Flexibilität in der Einbindung von verschiedenem Ausgangsmaterial wie Mono, Stereo, 5.1-Surround-Aufnahmen und B-Format ist im Beispielprojekt von Bedeutung.

Die Bewegungssteuerung ist ein zentraler Aspekt, da eine flexible Steuerung der Positionsparameter für die verschiedenen Arten von Bewegungen gebraucht wird. Dazu müssen sowohl die Positionsparameter wie Azimut und Elevation als auch eine Kontrolle über die wahrgenommene Distanz des Elements möglich sein. Diese Parameter sollen sowohl in Ableton automatisierbar als auch von externen Modulatoren wie den Ableton-nativen Geräten LFO oder Envelop Follower gesteuert werden können.

Eine zentrale Anforderung stellt außerdem die Kontrolle über die verwendete Raumsimulation dar. Vor allem beim Arpeggio, das sich in der Distanz zum Hörer verändern soll, spielt eine dynamische Anpassung von Early Reflections und Nachhall eine wichtige Rolle. Zudem ist eine detaillierte Kontrolle über die Eigenschaften der verwendeten Raumsimulation relevant, da es viele gestalterische Möglichkeiten bietet.

Für Elemente, die gleichzeitig bewegt werden sollen, ist es außerdem wichtig, dass eine Möglichkeit besteht, mehrere Elemente als Gruppe zu bewegen.

Ebenfalls soll das in Kapitel 3.4 beschriebene Headtracking-System in das Beispielprojekt integriert werden.

Da die Effizienz des Workflows einen großen Einfluss auf die praktische Nutzung der Ansätze hat, wurde als weiteres Kriterium der Einrichtungsaufwand und die Integration in Ableton Live aufgenommen.

Diese Anforderungen liefern die Kriterien, anhand derer die Ansätze schließlich verglichen werden sollen. Sie liefern keine vollständige Beurteilung über die Qualität der jeweiligen Softwarelösungen, sondern sollen die verschiedenen Möglichkeiten und Einschränkungen bei der Umsetzung des Beispielprojekts veranschaulichen.

6. Anwendung im Beispielprojekt

6.1 Envelop for Live

Integration in Ableton Live

E4L ließ sich in das bestehende Projekt einbinden, indem der *E4L Master Bus* auf eine leere Audiospur geladen wurde. In diesem Projekt wurde der modifizierte *Master Bus for Headtracking* verwendet. Die Headtracking-Daten wurden per OSC an den Master Bus übertragen. Als Decoder wurde das Preset „Binaural IEM“ gewählt, worüber das VST „Binaural Decoder“ der IEM-Plugin-Suite zum Dekodieren des Ambisonics-Signals auf Kopfhörer eingebunden wurde. Die verwendeten HRTFs hängen vom gewählten Preset ab. Im Falle des IEM Binaural Decoder waren es die Neumann KU 100 HRTFs.

Durch das Laden eines E4L Panner-Geräts, wie etwa des *Source-Panners*, wird automatisch das Routing der Spur auf „Sends Only“ gesetzt. Dadurch wird verhindert, dass das ursprüngliche Stereosignal zusätzlich an die Ableton-Masterspur weitergegeben wird, was zu einer Dopplung des mit dem Ambisonics-Signal führen würde. Außerdem wurde durch das Laden von weiteren E4L-Geräten in einer Kette automatisch das Routing der Devices geändert.

Bewegungssteuerung

Die Steuerung der Bewegung einzelner Quellen erfolgte über die Parameter Azimut, Elevation und Radius auf der Benutzeroberfläche der Panner-Geräte. Einige der E4L-Geräte verfügen bereits über integrierte LFOs für die Modulation dieser Parameter, die das schnelle Umsetzen grundlegender Bewegungsmuster, wie kontinuierliche Rotationen oder periodischer Schwingungen ermöglichten.

Im Beispielprojekt wurden komplexere Bewegungsabläufe durch eine Kombination von manueller Automation und Modulatoren wie dem *LFO* oder *Envelope Follower* aus Ableton Live gesteuert.

Da alle Parameter wie gewohnt über Automationen, Envelopes und Modulatoren angesprochen werden konnten, unterschied sich der Workflow kaum von einem herkömmlichen Stereo-Projekt. Einfache Bewegungen wurden mit den integrierten LFOs umgesetzt, für komplexere Abläufe wie abrupte Sprünge oder Fly-Overs waren manuelle Automationen aber besser geeignet.

Unterschiedliche Quellformate

Im Beispielprojekt sollten neben Mono- und Stereomaterial auch Field-Recordings im Ambisonics-B-Format sowie 5.1-Surround-Aufnahmen eingebunden werden.

Für Mono- und Stereomaterial reichte es aus, einen der E4L-Panner auf die jeweilige Audiospur zu laden. Das Routing erfolgte automatisch, was den Workflow unkompliziert machte.

Zum Einbinden von B-Format-Aufnahmen wurde der *E4L B-Format-Sampler* genutzt. Damit kann eine B-Format-WAV-Datei bis dritter Ordnung per Drag & Drop geladen werden, um diese in den E4L-Signalfloss einzuspeisen. Die Bedienung des Samplers beschränkt sich auf grundlegende Funktionen wie Start, Stopp und Pause. Das Fehlen von Funktionen wie dem Festlegen einer Startposition oder einer Loop-Region machte es schwierig, die Aufnahme im Kontext des Soundscapes gezielt einzusetzen und zeitlich mit dem restlichen Material abzustimmen.

Mehrkanaldateien wie 5.1-Mischungen erfordern einen Workaround, da Ableton Live keine Mehrkanalformate nativ unterstützt. Deshalb mussten die Dateien vorab in Einzelkanäle aufgeteilt werden, was den Workflow innerhalb von Ableton etwas aufbrach. In Ableton werden diese anschließend auf einzelne Audiospuren verteilt und gruppiert. Auf die Gruppenspur wurde ein *E4L Multi-Panner* geladen, der bis zu 8 Stereopaare als Eingänge bietet. Durch das entsprechende Routing lassen sich die Monospuren auf die Eingangskanäle des *Multi-Panners* verteilen und innerhalb eines einzigen Geräts manuell positionieren. Abbildung 18 zeigt die Benutzeroberfläche des *Multi-Panners* mit fünf aktiven Eingangskanälen im Beispielprojekt. Die einzelnen Kanäle wurden darin entsprechend ihrer Funktion in einem typischen 5.1-Setup positioniert, der LFE-Kanal blieb ungenutzt. Durch die globalen Parameter des *Multi-Panners* lassen sich zudem alle Kanäle gemeinsam bewegen.

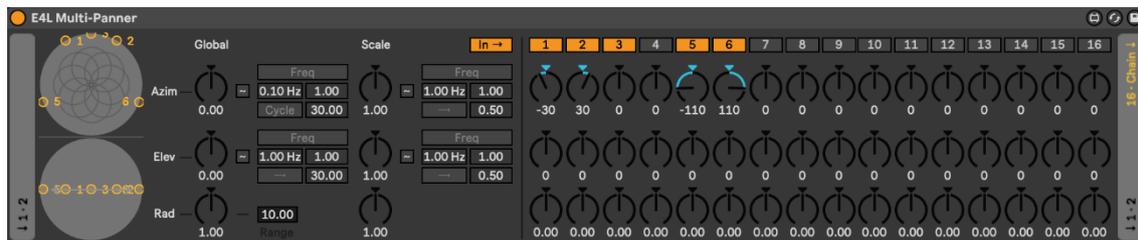


Abbildung 18: Der E4L Multi-Panner mit fünf aktiven Eingangskanälen

Gruppierung mehrerer Quellen

Es sollten mehrere Elemente des Beispielprojekts gemeinsam für eine Rotation um den Hörer bewegt werden. Dafür wurde ebenfalls der *E4L Multi-Panner* genutzt.

Die betroffenen Spuren wurden dazu vor der Encodierung auf den Multi-Panner geroutet, wo sie dann einzeln positioniert und als Gruppe weiterverarbeitet wurden. Das ermöglichte zwar die einzelnen Eingänge sowohl unabhängig voneinander als auch als Gruppe zu bewegen, verhinderte aber das weitere Verarbeiten der individuellen Quellen. Alle eingehenden Signale wurden nach dem Panning in ein einzelnes 3rd-Order-Ambisonics Signal kodiert.

Dadurch konnten die einzelnen Elemente danach nicht mehr individuell bearbeitet werden, um beispielsweise verschiedene Reverb- oder Delay-Geräte einzusetzen. Das war bei Signalen wie den Field Recordings unproblematisch, da sie im Beispielprojekt ohnehin ausschließlich gemeinsam bewegt und nicht weiter individuell verarbeitet wurden. Für Elemente, die nur für einzelne Bewegungen gruppiert werden sollten, war die Methode mit dem Multi-Panner allerdings zu unflexibel, da für sie auch individuelle Bearbeitung geplant war.

Neben der Gruppierung für Bewegungen wurden im Beispielprojekt auch die Möglichkeiten zur Erstellung von Subgruppen-Spuren zur Verarbeitung mehrerer Signalketten vor dem *E4L Master Bus* genutzt. Dazu wurden mit *E4L Aux Send*-Geräten Signale von mehreren Signalketten auf einer separaten Audiospur empfangen. Auf der Subgruppen-Spur wurde anschließend ein Multiband-Kompressor eingebunden, der HOA-Signale verarbeiten kann. Über ein *E4L 30A Plugin Send*-Gerät wurde das Signal an das externe Plugin übergeben und anschließend wieder über ein *30A Plugin Return*-Gerät zurück in den Signalfluss von E4L gebracht. Abbildung 8 zeigt die Geräte-Kette auf der Subgruppen-Spur. Die Einbindung von Drittanbieter-Plugins funktionierte größtenteils reibungslos, in seltenen Fällen kam es zu Routingfehlern, die sich aber durch ein Neustarten des Projekts beheben ließen. Das Integrieren von Instrumenten, die Ambisonics dritter Ordnung ausgeben, war nicht vorgesehen, ist aber generell möglich.

Kontrolle über Raumsimulation

Für das Erzeugen von Reflexionen und Nachhall stand das Gerät *E4L B-Format Convolution Reverb* zur Verfügung. Die Möglichkeit beliebige Impulsantworten zu laden und diese zu skalieren, bot viele Optionen zur Raumgestaltung. So konnten die Raumgröße und die Atmosphäre des Soundscapes aus einer gestalterischen Perspektive ausreichend beeinflusst werden.

Für die Steuerung der wahrgenommenen Distanz einzelner Elemente kam eine Kombination aus *Source-Panner* und dem *E4L B-Format Convolution Reverb* zum Einsatz. Über die aktivierte Distance-Lock-Funktion ließ sich der Eindruck von Distanz zum Hörer durch Abschwächung des Signals und Anpassung der ER und des Nachhalls steuern.

6.2 XP4L

Integration in Ableton Live

Um XP4L in das Beispielprojekt zu integrieren, wurde zuerst *xp.visual* auf eine leere Audiospur geladen, um die XP-Anwendung zu starten. Dann wurde in den Listener-Modus gewechselt, um das Headtracking zu aktivieren. Die OSC-Nachrichten wurden in der Reihenfolge Pitch-Yaw-Roll (PYR) übertragen und auf einem konfigurierbaren Port mit der Adresse */visual/listener/rotatexyz* empfangen. Über *xp.engine* wurde ein Dummy-Speaker-Layout und der Ausgabemodus Binaural gewählt. Bei XP4L werden die Kemar HRTFs benutzt, das Importieren von eigenen SOFA-Dateien ist hier nicht möglich.

Für jede Quelle wurde ein *xp.source* Gerät auf die entsprechenden Audiospuren geladen. Das kann vor allem bei größeren Projekten zeitaufwendig sein. Bei der Verwendung beider Kanäle einer Stereospur müssen die Kanäle zudem manuell auf zwei *xp.source* Devices aufgeteilt und separat positioniert werden. Die Benennung der Spuren wird automatisch beim Laden von XP4L-Geräten angepasst, da die interne Signalverarbeitung von XP4L darauf basiert. Deshalb ist eine nachträgliche Bearbeitung der Spurnamen nicht mehr möglich. Das führt dazu, dass Projekte mit vielen Sources schnell unübersichtlich werden können.

In allen Geräten lassen sich Presets der aktuellen Einstellungen sichern, was vor allem bei den *xp.room*-Geräten sehr nützlich war, um schnell zwischen verschiedenen Raumeinstellungen zu wechseln.

Die Solo-Funktion von Ableton war bei der Nutzung von XP4L weniger praktikabel, da immer sowohl die entsprechende Quelle als auch *xp.engine* auf Solo geschaltet werden mussten, um eine Quelle einzeln abzuhören. Das führte vor allem bei größeren Projekten zu einem weniger flüssigen Workflow.

Bewegungssteuerung

Um verschiedene Arten von Bewegungen zu realisieren, wurden die Positionsparameter von *xp.source* verwendet. Für Azimut, Elevation und Distanz gibt es jeweils einen integrierten LFO, mit dem einfache Bewegungen umgesetzt werden können. Allerdings lässt sich die Modulationstiefe nicht anpassen. Für komplexere Bewegungsmuster wurden daher externe LFOs oder Automationen verwendet.

Im Beispielprojekt sollte bei einem Synthesizer-Arpeggio langsam der Azimut-Parameter moduliert werden, sodass eine kontinuierliche Rotation um den Hörer entsteht. Gleichzeitig sollte die Distanz sich vergrößern und dann wieder auf den ursprünglichen Wert zurückkehren. Für die Azimut-Automation war hier der interne LFO völlig ausreichend. Für die Distanzautomation

wurde an dieser Stelle ein externer LFO benötigt, da die Modulationstiefe gezielt eingestellt werden musste.

Ein weiteres Gerät zur Bewegung von Quellen in XP4L ist *xp.source.anime*. Es wird nach einem *xp.source*-Gerät geladen und kann zum Generieren verschiedener Bewegungsmuster genutzt werden. Während sich damit auch klassische Bewegungen umsetzen ließen, war es vor allem interessant für das Experimentieren mit generierten, unvorhersehbaren Flugbahnen. Für gezielte und geplante Bewegungen waren allerdings manuelle Automation und externe LFOs präziser.

Unterschiedliche Quellformate

In XP4L werden alle Quellen als Mono-Objekte repräsentiert. Es lassen sich zwar die einzelnen Kanäle von Stereo- und Mehrkanaldateien auf mehrere Instanzen von *xp.source* aufteilen, doch die Entscheidung zur Nutzung aller Kanäle ist eher abhängig vom gewünschten räumlichen Effekt. Im Beispielprojekt wurden für Signale wie Pads und Atmosphären, die ausgedehnt wahrgenommen werden sollten, beide Kanäle einer Stereoquelle über zwei *xp.source*-Objekte im Raum positioniert. Für Quellen, die genauer lokalisiert werden sollten und komplexere Bewegungen durchführten, wurde meist nur ein Kanal verwendet.

In den Fällen, in denen beide Kanäle eines Stereo-Signals benutzt wurden, erschwerte die Aufteilung der Kanäle auf zwei unabhängige Objekte die Arbeit bei Positionierung und Automation der Objekte als Paar.

Ein 5.1-Surround Field-Recording wurde mit dem Gerät *xp.sm* in einzelne Kanäle aufgeteilt und jeweils über ein *xp.source*-Objekt platziert, um eine kohärente Grundatmosphäre zu schaffen. Das direkte Einbinden von B-Format-Ambisonics Field-Recordings ist in XP4L nicht vorgesehen.

Gruppierung mehrerer Quellen

Um mehrere Elemente gleichzeitig zu bewegen, wurde das *xp.group*-Gerät genutzt. Die Sources, die gemeinsam bewegt werden sollten, wurden einer bestimmten Gruppe zugewiesen. Die Bewegungen der Gruppe wurden über Automationen gesteuert, beispielsweise als Rotation um die Z-Achse. Die relative Position der Quellen zueinander bleibt dabei gleich. Das machte es möglich, Anordnungen von Quellen aufzubauen und diese als Einheit zu bewegen. So wurde beispielsweise ein Delay-Effekt für ein Element umgesetzt, indem mehrere Return-Spuren mit unterschiedlichen Delay-Einstellungen um das Element herum positioniert wurden. Diese konnten dann alle einer Gruppe zugewiesen und gemeinsam bewegt werden.

Das Zuweisen von verschiedenen Elementen zu einer Gruppe war auch eine Lösung, auf verschiedene Objekte aufgeteilte Stereopaare zu positionieren und zu automatisieren. Durch die Begrenzung auf maximal fünf verschiedene Gruppen war es aber notwendig, sich dabei auf die für das Projekt wichtigsten Konstellationen zu beschränken.

Kontrolle über Raumsimulation

Die Benutzeroberfläche von *xp.room* bietet Zugang zu den zentralen Parametern der zugrundeliegenden *spat~* Reverb-Engine und erlaubt damit eine tiefreichende Anpassung der Raumsimulation. Im Beispielprojekt wurden verschiedene Räume mit unterschiedlichen Raumgrößen und Decay-Zeiten erstellt. Für perkussive und transientenreiche Signale wurden kürzere Decay-Zeiten verwendet, während flächigere Signale einem größeren Raum mit längerem Nachhall zugewiesen wurden. Über die *Perceptual Factors* der einzelnen Quellen konnten zudem weitere Einstellungen wie *Room Presence* angepasst werden. Die Raumzuweisung konnte pro Quelle direkt über *xp.source* durchgeführt werden, sodass kein zusätzliches Routing notwendig war. Das machte ein schnelles Vergleichen von verschiedenen Raumeinstellungen möglich, indem nacheinander durch die Raumzuweisung eines Objekts geschalten wurde.

6.3 SPAT Revolution

Setup und Integration in Ableton Live

Um SPAT Revolution in Ableton Live zu integrieren, wurde der virtuelle Audiotreiber BlackHole (*BlackHole*, 2019) verwendet. Dieser wurde in Ableton als Ausgangs- und in SPAT als Eingangsgerät gewählt und bietet bis zu 64 Ein- und Ausgangskanäle. Über die „Audio To“-Sektion (siehe Abbildung 19) konnte das Routing jeder Spur auf einzelne Kanäle von BlackHole festgelegt werden, die in SPAT von Input-Modulen mit entsprechender Konfiguration aufgegriffen wurden.



Abbildung 19: Routing der Ableton-Spuren auf die Kanäle von BlackHole

Für jede Quelle wurde ein *SPAT Send*-Plugin auf der entsprechenden Ableton-Spur für die Steuerung der Source-Parameter aus Ableton heraus geladen und für jedes *Room-Modul* eine Spur mit einem *SPAT Room*-Plugin erstellt. Die OSC-Verbindungen mussten dafür in den SPAT-Einstellungen aktiviert werden.

Auch für die Headtracking-Integration musste zunächst in den SPAT-Einstellungen eine benutzerdefinierte Input-OSC-Connection erstellt werden. Dieser konnte ein frei wählbarer Port zugewiesen werden und über die OSC-Adresse `/room/k/ypr` konnte in der Reihenfolge Yaw, Pitch, Roll (YPR) die Listener-Rotation im jeweiligen Raum übertragen werden, wobei k hier den Raumindex bezeichnet. Dadurch ist es möglich, Headtracking-Daten auch an mehrere Räume gleichzeitig zu senden, um so mehrere Hörer parallel zu tracken.

Zusätzlich wurde in SPAT ein weiteres Stereo *Input-Modul* für Cue- und Master-Out von Ableton verwendet und direkt mit dem Output-Modul verbunden, um die Vorhörfunktion weiter nutzen zu können.

Bewegungssteuerung

Die Bewegungen einzelner Elemente wurden durch die Kontrolle der Azimut-, Elevation- und Distance-Parameter der *SPAT Send*-Plugins realisiert. In SPAT waren zwar keine integrierten Modulatoren vorhanden, die Parameter der Plugins konnten aber alle automatisiert und moduliert werden, was die Kontrolle innerhalb von Ableton ermöglichte.

Durch die visuelle Darstellung der Objekte in der SPAT-Anwendung konnten komplexere Bewegungsmuster kontrolliert werden.

Unterschiedliche Quellformate

Für die Integration von Mono- und Stereoquellen wurden die Ausgänge der entsprechenden Ableton-Spuren über BlackHole geleitet. In SPAT wurde dann ein passendes Input-Modul in Mono- oder Stereokonfiguration erstellt.

Für andere Quellformate wurde beispielsweise ein Input-Modul mit 5.0-Konfiguration erstellt. Die Mehrkanaldateien mussten außerhalb von Ableton aufgeteilt werden, und auf separate Kanäle geleitet werden, die dann vom *Input-Modul* aufgegriffen werden konnten. In gleicher Weise war es möglich, eine Audiodatei im B-Format auf einzelne Kanäle aufzuteilen und diese über ein *Input-Modul* mit passender Konfiguration sowie einen *Input-Transcoder* an den *Binaural-Room* zu übergeben.

Gruppierung mehrerer Quellen

Eine Gruppierung mehrerer Quellen zur gemeinsamen Bewegung war im Rahmen des Beispielprojekts nicht möglich, da SPAT keine native Gruppierung für Sources innerhalb eines Raumes bietet. Dadurch mussten Bewegungen, die mehrere Elemente gleichzeitig betreffen sollten, für jede Quelle separat umgesetzt werden. Das erhöhte den Aufwand, der für Korrekturen notwendig war deutlich, da alle Änderungen einzeln durchgeführt werden mussten. Das konnte zwar durch Workarounds umgangen werden, etwa durch das Zuweisen von synchronisierten LFOs auf Parameter von mehreren *Spat Send*-Instanzen oder dem Kopieren und manuellen Anpassen von Automationen auf mehrere Spuren, führte aber durch die Verkomplizierung des Prozesses zu einem weniger intuitiven Workflow und beeinträchtigte die spontane Umsetzung von Ideen deutlich.

Dafür wurden aber Quellen, die mit mehreren Eingangskanälen erstellt wurden in ihrer jeweiligen Konfiguration als Einheit betrachtet. So konnten etwa Stereopaare gemeinsam bewegt werden.

Kontrolle über Raumsimulation

Im Beispielprojekt wurden die Raumeinstellungen über die *SPAT Room*-Plugins innerhalb von Ableton gesteuert. Dort waren alle Parameter der *spat~* Reverb-Engine verfügbar. Es wurden im Beispielprojekt verschiedene *Room-Module* benutzt, die sich in Einstellungen und Ausgabebetyp (Stereo, Binaural) unterscheiden. Zudem wurde ein Raum mit dem Panning-Typ Nearfield-Binaural angelegt und für Elemente verwendet, die sich sehr nah an den Hörer bewegen sollten. Die verwendeten HRTFs konnten für jeden Raum gewählt werden und es gab die Möglichkeit eigene SOFA-Dateien zu importieren.

In SPAT bestand die Möglichkeit Presets für Raumeinstellungen zu sichern und zu laden, außerdem waren eine Reihe an vorgefertigten Presets verfügbar. Das erleichterte den Workflow erheblich, da schnell verschiedene Raumeindrücke getestet und angepasst werden konnten.

7. Vergleich und Diskussion

Die drei getesteten Softwarelösungen werden in diesem Kapitel anhand der in Kapitel 5.2 definierten Kriterien verglichen. Dabei werden die in der Anwendung im Beispielprojekt gezeigten Stärken und Schwächen zusammengefasst und diskutiert.

7.1 Setup und Integration in Ableton

Für die Einrichtung und Integration der verschiedenen Ansätze in Ableton Live war der Aufwand unterschiedlich hoch. Im Beispielprojekt war aber nicht nur der Einrichtungsaufwand wichtig, sondern auch wie stark die gewohnte Arbeitsweise in Ableton verändert wurde.

Envelop for Live ließ sich am einfachsten in Ableton Live integrieren. Durch das automatische Routing der E4L-Geräte und den Zugriff auf alle Bedienelemente, Visualisierungen und Signalwege direkt innerhalb der DAW unterschied sich der Workflow kaum von dem eines herkömmlichen Stereo-Projekts.

Der Einrichtungsaufwand von XP4L entsprach zunächst etwa dem von E4L. Dadurch, dass das Routing automatisch beim Laden der Devices geändert wurde, konnten Quellen schnell integriert und positioniert werden. Die Möglichkeit in fast allen Geräten Presets der aktuellen Einstellungen abzuspeichern und zu laden, erleichterte das Wiederaufrufen und Vergleichen von Einstellungen. Durch die automatische Umbenennung der Spuren durch die XP4L-Geräte wurden größere Projekte allerdings schnell unübersichtlich, vor allem weil zur visuellen Kontrolle erst in die XP-Anwendung gewechselt werden musste.

SPAT Revolution brachte mit der Einrichtung des Signalflusses in der SPAT-Anwendung und dem manuellen Routing aller Spuren über BlackHole den größten Aufwand mit sich. Zusätzlich mussten für jede Spur die SPAT-Plugins geladen werden. Auch wenn das Setup aufwendiger war, konnte danach vergleichbar mit Envelop for Live weitergearbeitet werden. Die Erstellung neuer Quellen brachte aber immer einen weiteren Arbeitsschritt mit sich, da dafür zusätzlich in die SPAT-Anwendung gewechselt werden musste.

Ein weiterer Unterschied besteht darin, dass im Gegensatz zur Begrenzung von XP4L auf bis zu vier Räume und 32 Sources, bei Envelop For Live und SPAT Revolution keine direkten Begrenzungen vorhanden sind. Das macht diese Lösungen insbesondere bei Projekten mit einer sehr großen Anzahl an Quellen interessant.

Das Wechseln der verwendeten HRTFs und das Importieren eigener SOFA-Dateien ist bei SPAT Revolution ohne Probleme möglich. In Envelop 4 Life werden die verwendeten HRTFs vom ausgewählten Decoder-Preset bestimmt, lassen sich aber durch das Einbinden eigener Decoder

oder das Anpassen der zugrundeliegenden Max-Patches austauschen. Bei XP4L waren hingegen nur die Kemar-HRTFs verfügbar, was ein Tauschen oder Individualisieren verhinderte.

7.2 Bewegungssteuerung

Die unkomplizierte und präzise Umsetzung von Bewegungen einzelner Elemente stellt eine zentrale Anforderung an einen Workflow für binaurale Produktionen in Ableton dar. Um die Umsetzung komplexer Bewegungsmuster zugänglich zu machen, sind die Aufnahme von Automationen, Visualisierungen und detaillierte Einstellungen von Modulatoren wichtig.

Envelop For Live bot das Nutzen von integrierten LFOs in verschiedenen Panner- und Effektgeräten an. Diese waren ausreichend konfigurierbar und ermöglichten das schnelle Umsetzen von kontinuierlichen Bewegungen. Durch das bei jedem Panner-Gerät vorhandene visuelle Feedback war das Einstellen von komplexeren Bewegungen innerhalb eines Geräts möglich.

In XP4L waren ebenfalls interne LFOs verfügbar, mit denen sich Azimut, Elevation und Distance modulieren ließen. Bei diesen konnte aber nicht die Modulationstiefe angepasst werden, was sie für die Umsetzung von vielen Bewegungsmustern unbrauchbar machte. Zusätzlich waren mit den XP4L-Geräten *xp.source.simply* und *xp.pan* zwei Möglichkeiten verfügbar, Bewegungen innerhalb von Ableton zu visualisieren und über Steuerung per Maus zu automatisieren. Für eine experimentellere Methode eignete sich *xp.source.anime* gut, da es unvorhersehbare Effekte erzeugte, lies sich aber auch für einfachere Bewegungen benutzen. Innerhalb des Beispielprojekts wurden allerdings ausschließlich externe LFOs und manuelle Automationen der einzelnen Parameter verwendet, da sich diese Vorgehensweise als am präzisesten erwies.

Da SPAT Revolution ermöglichte die Bewegung einzelner Objekte nur über die Parameter der *SPAT Send*-Plugins, bot also keine internen LFOs. Die Bewegung war trotzdem über manuelle Automationen der einzelnen Parameter oder über externe LFOs möglich.

Insgesamt waren bei allen Ansätzen die Parameter für Bewegungen frei über die in Ableton verfügbaren Möglichkeiten manipulierbar. Die konfigurierbaren LFOs von E4L erleichterten das schnelle Umsetzen von Ideen, grundsätzlich waren komplexe Bewegungen aber in allen Ansätzen über die Automationsmöglichkeiten von Ableton umsetzbar.

7.3 Unterschiedliche Quellformate

Das Einbinden von verschiedenen Quellformaten war eine Anforderung für das Beispielprojekt, da Teil des vorhandenen Audiomaterials unter anderem aus 5.1-Surround-Aufnahmen und Field Recordings im B-Format bestand. Es sollte getestet werden, wie sich solche Formate in die jeweiligen Systeme integrieren lassen. Auch wenn es grundsätzlich möglich ist, etwa 5.1-Dateien als einzelne Monospuren zu behandeln, kann die vorhandene Unterstützung für solche Formate oder etwaige Workarounds den Workflow verbessern und zeigen, wie flexibel ein System auf unterschiedliches Ausgangsmaterial reagiert.

In Envelop for Live bestand ein Workaround für das Integrieren von die 5.1-Surround-Aufnahmen darin, alle Kanäle einzeln auf Audiospuren zu importieren und diese auf einzelne Kanäle eines Multi-Panners zu routen. Dort konnten sie einzeln positioniert und über die globalen Einstellungen als Gruppe bewegt werden. Das erwies sich als funktionale Lösung, da diese Methode im Vergleich zur Verarbeitung aller Kanäle durch einzelne Panner übersichtlicher war. Für die Field-Recordings im B-Format stand ein Sampler zur Verfügung, der das unkomplizierte Einbinden per Drag & Drop ermöglichte. Dadurch konnten die Aufnahmen ohne vorheriges Setup abgespielt werden. Durch den Mangel an Möglichkeiten des Samplers war es aber umständlich, die Aufnahmen tatsächlich im Gestaltungsprozess zu verwenden.

XP4L ermöglichte mit dem Gerät xp.sm, Mehrkanaldateien in Einzelkanäle zu teilen. Diese konnten dann einzeln mit je einem Source-Gerät positioniert werden. Das Einbinden der Aufnahmen im B-Format war nicht unterstützt.

Durch die Möglichkeit der Erstellung von *Input-Modulen* mit unterschiedlicher Konfiguration in SPAT, konnten sowohl Stereo, 5.1- als auch die B-Format-Aufnahmen integriert werden. Die Kanäle wurden auch hier einzeln an SPAT geleitet, und dort vom jeweiligen Modul aufgefasst und gegebenenfalls von *Input-Transcoder-Modulen* ins richtige Format gebracht.

Da XP4L ausschließlich mit Mono-Objekten arbeitet, musste bei allen Quellen entschieden werden, welche der Kanäle genutzt werden sollen. Bei E4L und SPAT war es dahingegen möglich, beispielsweise Stereo-Dateien als zusammengehörig zu behandeln, ohne zwei separate Objekte erstellen zu müssen. Das erleichterte den Umgang mit solchen Quellen. Im Beispielprojekt blieb es unabhängig davon auch eine gestalterische Frage. Je nach gewünschtem Räumlichen Eindruck wurde entschieden, ob alle oder nur einzelne Kanäle eines Mehrkanalformats genutzt wurden.

7.4 Gruppierung mehrerer Elemente

Die Möglichkeit mehrere Elemente zu gruppieren und gemeinsam zu bewegen war besonders bei räumlich oder inhaltlich zusammenhängenden Elementen relevant. Dadurch konnte deren Konstellation während der Bewegung beibehalten werden und es ersparte das Bewegen jeder einzelnen Quelle. Das vereinfachte den Workflow erheblich.

Der *Multi-Panner* in E4L machte zwar die Bewegung von Gruppierungen möglich, hatte aber den Nachteil, dass danach keine Einzelverarbeitung mehr möglich war.

Dahingegen war es in XP4L direkt möglich, beliebig viele Quellen einer Gruppe zuzuweisen, um diese gemeinsam zu bewegen. Durch die Beschränkung auf fünf Gruppen musste man aber klar auswählen, wie man diese zuweist.

SPAT hingegen besaß keine native Möglichkeit mehrerer Objekte gleichzeitig aus Ableton heraus zu steuern. Es konnten zwar durch Workarounds wie durch synchronisierte Modulation ähnliche Ergebnisse erzielt werden, war aber deutlich aufwendiger und weniger intuitiv.

7.5 Kontrolle über Raumsimulation

Die Kontrolle über Raumsimulation und Nachhall spielte im Beispielprojekt eine wichtige Rolle, da sie nicht nur für gestalterische Zwecke genutzt werden kann, sondern auch entscheidend zur korrekten Lokalisierung und Externalisierung beiträgt (Begault et al., 2001, S. 7).

E4L bot mit dem *E4L B-Format Convolution Reverb* eine einfache Möglichkeit zur Simulation von Early-Reflections und Nachhall. Die Auswahl und Modifikation verschiedener Impulsantworten ermöglichten die Anpassung des Raumklangs auf die jeweilige Anwendung. Im Vergleich zu den Möglichkeiten der in XP4L und SPAT verwendeten *spat~*-Engine ist die Kontrolle über einzelne akustische Eigenschaften jedoch eingeschränkt.

Durch die Nutzung der *spat~*-Engine ließ sich in XP4L und SPAT eine detaillierte Kontrolle über Raumeigenschaften erreichen. Dabei konnten Early-Reflections, Cluster und Nachhall genau angepasst werden und mit den Perceptual Factors über ein benutzerfreundliches Interface gesteuert werden. Zusätzlich bot SPAT eine Auswahl an Presets, die ein schnelles Vergleichen von vorgefertigten Raumeinstellungen ermöglichten.

7.7 Headtracking Integration

Da Headtracking das Auftreten von Front-Back Confusions und In-Kopf-Lokalisationen verringern kann, ist es bei Workflows zur Produktion von binauralen Inhalten von großer Bedeutung. Deshalb war eine der Anforderungen, Headtracking in 3 DOF zu integrieren.

In allen Ansätzen war das Benutzen eines Headtrackers beim Monitoring umsetzbar. XP4L und SPAT Revolution hatten dabei eine offizielle Unterstützung, während bei Envelop For Live auf ein durch die Community bereitgestelltes modifiziertes Gerät zurückgegriffen wurde.

8. Fazit

Das Beispielprojekt stellte konkrete Anwendungsszenarien dar, aus denen sich die Anforderungen an die verschiedenen Ansätze ergaben. Das präzise Steuern von Bewegungen, der Umgang mit Ausgangsmaterial in verschiedenen Quellformaten, die Gruppierung von Elementen zur gemeinsamen Bewegung sowie die Kontrolle über die Eigenschaften der verwendeten Raumsimulation sind für einen Workflow zur Erstellung von binauralen Inhalten in Ableton Live entscheidend.

Envelop 4 Live bietet von den Lösungen die schnellste Integration und schafft einen nahtlosen Workflow, der eine intuitive Arbeitsweise ermöglicht. SPAT ist dagegen aufgrund der externen Routingstruktur setupintensiver. XP4L ist zwar effektiv in Ableton eingebunden, verliert aber an Übersicht bei großen Projekten. Alle Ansätze machen die Bewegungssteuerung voll über Automation und Modulation zugänglich, wobei E4L durch integrierte LFOs eine besonders schnelle Umsetzung einfacher Bewegungsideen ermöglicht. XP4L dagegen bietet unkonventionelle Möglichkeiten zur Erzeugung von Bewegungsmustern und die Steuerung von Gruppen. Während XP4L auf Mono-Quellen basiert, bieten E4L und SPAT geeignete Möglichkeiten zum Umgang mit Ausgangsmaterial in verschiedenen Formaten. Alle Ansätze verfügen außerdem über die Möglichkeit einer Raumsimulation, wobei XP4L und SPAT mit dem algorithmischen Reverb von IRCAM *spat*~ eine umfassend-konfigurierbare Simulation bieten. Um eine Übersicht über die Einordnung der drei Varianten anhand der Anforderungen des Beispielprojekts zu geben, werden die Ergebnisse in der folgenden Tabelle zusammengefasst.

Tabelle 1: Vergleich der getesteten Ansätze anhand der Kriterien. Legende: ++ = sehr gut erfüllt, + = gut erfüllt, 0 = eingeschränkt erfüllt, - = nicht zufriedenstellend erfüllt. (eigene Darstellung)

<i>Kriterium</i>	<i>Envelop For Live</i>	<i>XP4L</i>	<i>SPAT Revolution</i>
<i>Setup und Integration</i>	++	+	0
<i>Bewegungssteuerung</i>	++	++	+
<i>Unterschiedliche Quellformate</i>	+	0	++
<i>Gruppierung von Quellen</i>	0	++	-
<i>Kontrolle über Raumsimulation</i>	+	++	++
<i>Headtracking-Integration</i>	+	++	++

Jede der drei getesteten Softwarelösungen zeigte Schwächen und Stärken je nachdem, was umgesetzt werden sollte und welche Produktionsphase im Fokus stand. Während Envelop for Live Ambisonics dritter Ordnung in Ableton integriert, verfolgen XP4L und SPAT Revolution einen objektbasierten Ansatz auf der Grundlage von *spat~*. Auch wenn XP4L und SPAT Revolution die gleiche Grundlage nutzen, gestalten sie die Benutzeroberfläche und den Workflow sehr unterschiedlich.

E4L zeichnete sich durch die schnelle, unkomplizierte Integration, die Möglichkeit des Einsatzes von Ambisonics-basierten Effekten und die reibungslose Integration in den Produktionsablauf aus. Besonders in kreativen Produktionsphasen, wie bei schnellen Skizzen und Sounddesign-Prozessen, bei denen Bedienbarkeit und Tempo im Fokus stehen, hatte E4L seine Stärken.

SPAT Revolution bietet hingegen ein hohes Maß an Flexibilität in Skalierung und Kompatibilität mit verschiedenen Formaten und Setups. Durch das modulare Routing-Setup lässt sich SPAT auf die unterschiedlichsten Produktionsszenarien anpassen und hat von den drei Ansätzen die meisten Konfigurationsmöglichkeiten. Im Gegensatz zu E4L oder XP4L ist SPAT aber keine speziell für Ableton Live entwickelte Software und hat dadurch spürbare Einschränkungen im Workflow, wie zum Beispiel das häufige Wechseln in die SPAT-Anwendung beim Erstellen von Objekten. Sobald die grundlegende Konfiguration abgeschlossen ist, lässt sich SPAT aber sinnvoll mit Ableton kombinieren und bietet den Zugriff auf die *spat~*-Engine.

XP4L beschreitet durch die Ableton-nahe Entwicklung und die *spat~* basierte Signalverarbeitung einen Mittelweg. Die Benutzeroberflächen der Max For Live-Geräte fügen sich gut in bestehende Ableton-Projekte ein und erlauben gleichzeitig die Kontrolle über räumliche Parameter inklusive Gruppen- und Raumzuweisungen.

Welche Variante in der Praxis die beste Lösung ist, hängt am Ende immer von den Anforderungen und Prioritäten des jeweiligen Projekts ab.

Abbildungsverzeichnis

Abbildung 1: Kopfbezogenes Kugelkoordinatensystem (Blauert & Braash, 2008, S.88)	3
Abbildung 2: Die Benutzeroberfläche von spat.oper mit den Perceptual Factors.....	10
Abbildung 3: Signalfluss für binaurale Wiedergabe in Envelop For Live (eigene Darstellung). 12	
Abbildung 4: Benutzeroberfläche des E4L Source Panners.....	12
Abbildung 5: Der E4L Source Panner im "Chain" Modus.....	13
Abbildung 6: Benutzeroberfläche des E4L Master Bus	13
Abbildung 7: Der Modifizierte E4L Master Bus mit Headtracking Integration	14
Abbildung 8: Eine Effektkette bestehend aus E4L Aux Return, E4L 30A Plugin Send/Return und IEM MultiBandCompressor	14
Abbildung 9: Benutzeroberfläche des E4L B-Format Convolution Reverbs.....	15
Abbildung 10: Signalfluss in XP4L bei binauraler Wiedergabe (eigene Darstellung).....	16
Abbildung 11: Benutzeroberfläche von xp.source in Standardeinstellung	17
Abbildung 12: Benutzeroberfläche von xp.engine.....	17
Abbildung 13: Benutzeroberfläche von xp.visual mit aktiviertem Headtracking	18
Abbildung 14: XP-Anwendung im Scene-Modus. Sichtbar sind drei Source-Objekte, positioniert durch xp.source	19
Abbildung 15: Setup-Ansicht in SPAT Revolution	20
Abbildung 16: Die automatisierbaren Parameter des SpatRevolution-Send Plugins.....	21
Abbildung 17: Der Signalfluss am Beispiel einer Mono-Quelle bei binauraler Wiedergabe in SPAT Revolution	22
Abbildung 18: Der E4L Multi-Panner mit fünf aktiven Eingagskanälen.....	26
Abbildung 19: Routing der Ableton-Spuren auf die Kanäle von BlackHole	31

Literaturverzeichnis

Asperti, A., Colasuonno, G., & Guerra, A. (2024). Illumination and Shadows in Head Rotation: Experiments with Denoising Diffusion Models. *Electronics*, *13*(15), Article 15.

<https://doi.org/10.3390/electronics13153091>

Begault, D., Wenzel, E., & Anderson, M. (2001). Direct Comparison of the Impact of Head Tracking, Reverberation, and Individualized Head-Related Transfer Functions on the Spatial Perception of a Virtual Speech Source. *Journal of the Audio Engineering Society. Audio Engineering Society*, *49*, 904–916.

BlackHole (Version 0.6.1). (2019). [Software]. Existential Audio Inc.

<https://existential.audio/blackhole/>

Blauert, J., & Braasch, J. (2008). Räumliches Hören. In S. Weinzierl (Hrsg.), *Handbuch der Audiotechnik* (S. 87–121). Springer. https://doi.org/10.1007/978-3-540-34301-1_3

Brinkmann, F., & Pike, C. (2020). Binauraltechnik. In S. Weinzierl (Hrsg.), *Handbuch der Audiotechnik* (S. 1–23). Springer. https://doi.org/10.1007/978-3-662-60357-4_27-2

Carpentier, T., Noisternig, M., & Warusfel, O. (2015). Twenty Years of Ircam Spat: Looking Back, Looking Forward. *41st International Computer Music Conference (ICMC)*, 270–277. <https://hal.science/hal-01247594>

Craven, P. G., & Gerzon, M. A. (1977). *Coincident microphone simulation covering three dimensional space and yielding various directional outputs* (United States Patent No.

US4042779A). <https://patents.google.com/patent/US4042779A/en>

d5x610. (2023). *Headtracked Binaural E4L Mod Of Master Bus version 1.54* (Version 1.54) [Software]. <https://maxforlive.com/library/device/9643/headtracked-binaural-e4l-mod-of-master-bus>

Eargle, J. M. (1996). Transaural Recording Techniques. In J. M. Eargle (Hrsg.), *Handbook of Recording Engineering* (S. 141–146). Springer US. https://doi.org/10.1007/978-1-4684-9919-3_11

Envelop for Live (Version 11.3.21). (2018). [Software]. Envelop. <https://www.envelop.us/>

Externals | Cycling '74 Documentation. (o. J.). Abgerufen 16. April 2025, von <https://docs.cycling74.com/userguide/externals/>

Felderhoff, U., Mackensen, P., & Theile, G. (1998). Stabilität der Lokalisation bei verfälschter Reproduktion verschiedener Merkmale der binauralen Signale. *20. Tonmeistertagung Karlsruhe, 1998, Tagungsband*. 20. Tonmeistertagung Karlsruhe, 1998, Karlsruhe.

- Fellgett, P. B. (1974). Ambisonic reproduction of directionality in surround-sound systems. *Nature*, 252(5484), 534–538. <https://doi.org/10.1038/252534b0>
- Futuresource. (2024). *Futuresource Consumer Headphones Market*. Statista. <https://www.statista.com/statistics/236075/revenue-of-headphone-shipments-in-the-united-states/>
- Hendrickx, E., Stitt, P., Messonnier, J.-C., Lyzwa, J.-M., Katz, B. F., & de Boishéraud, C. (2017). Influence of head tracking on the externalization of speech stimuli for non-individualized binaural synthesis. *The Journal of the Acoustical Society of America*, 141(3), 2011–2023. <https://doi.org/10.1121/1.4978612>
- ICST. (o. J.). *Downloads: Ambisonics Externals for MaxMSP | ZHdK.ch*. Züricher Hochschule Der Künste. Abgerufen 16. April 2025, von <https://www.zhdk.ch/forschung/icst/software-downloads-5379/downloads-ambisonics-externals-for-maxmsp-5381>
- Krebber, W., Gierlich, H.-W., & Genuit, K. (2000). Auditory virtual environments: Basics and applications for interactive simulations. *Signal Processing*, 80(11), 2307–2322. [https://doi.org/10.1016/S0165-1684\(00\)00119-5](https://doi.org/10.1016/S0165-1684(00)00119-5)
- Mackensen, P. (2004). *Auditive Localization. Head movements, an additional cue in Localization* [Technische Universität Berlin]. <https://api-depositonce.tu-berlin.de/server/api/core/bitstreams/5fa72a18-5bf2-4a90-9a9e-20d8fb7ee9cb/content>
- Mackensen, P., Fruhmann, M., Thanner, M., Theile, G., Horbach, U., & Karamustafaoglu, A. (2000). Head Tracker-Based Auralization Systems: Additional Consideration of Vertical Head Movements. *Journal of The Audio Engineering Society*.
- Margulies, J. (2010). *Ableton Live 8 power! : The comprehensive guide*. Boston, MA : Course Technology PTR. <http://archive.org/details/abletonlive8powe0000marg>
- Max 5 Help and Documentation*. (o. J.). Abgerufen 15. April 2025, von <https://docs.cycling74.com/legacy/max5/vignettes/intro/docintro.html>
- Middlebrooks, J. (2015). Sound localization. *Handbook of clinical neurology*, 129C, 99–116. <https://doi.org/10.1016/B978-0-444-62630-1.00006-8>
- Møller, H. (1992). Fundamentals of binaural technology. *Applied Acoustics*, 36(3), 171–218. [https://doi.org/10.1016/0003-682X\(92\)90046-U](https://doi.org/10.1016/0003-682X(92)90046-U)
- Nachar, C., Zotter, F., Deleflie, E., & Sontacchi, A. (2011). *Ambix-Suggesting an Ambisonics Format*. Ambisonics Symposium.
- Rudzki, T., & Majer, J. (2019). *NVSONICS Head Tracker OSC Bridge* [Software]. NVSonic. <https://github.com/trsonic/nvsonic-head-tracker/releases>

- Schörkhuber, C., Zaunschirm, M., & Höldrich, R. (2018, März 11). *Binaural Rendering of Ambisonic Signals via Magnitude Least Squares*. DAGA. <https://arxiv.org/abs/2501.18224>
- Slee, M. (2024). *Envelop For Live Wiki*.
<https://github.com/EnvelopSound/EnvelopForLive/wiki>
- Thurlow, W. R., & Runge, P. S. (1967). Effect of Induced Head Movements on Localization of Direction of Sounds. *The Journal of the Acoustical Society of America*, 42(2), 480–488.
<https://doi.org/10.1121/1.1910604>
- Wenzel, E. M. (1998). The impact of system latency on dynamic performance in virtual acoustic environments. *The Journal of the Acoustical Society of America*, 103(5_Supplement), 3026–3026. <https://doi.org/10.1121/1.422547>
- Wenzel, E. M. (2001). Effect of increasing system latency on localization of virtual sounds with short and long duration. *Proceedings of the 2001 International Conference on Auditory Display*, 185–190.
- XP for Ableton Live* (Version 1.30). (2021). [Software]. XP4L. <https://www.xp4l.com/>
- Zotter, F., & Frank, M. (2019). *Ambisonics: A Practical 3D Audio Theory for Recording, Studio Production, Sound Reinforcement, and Virtual Reality*. Springer Nature.
<https://doi.org/10.1007/978-3-030-17207-7>

Ehrenwörtliche Erklärung

Hiermit versichere ich, Timo Haas, ehrenwörtlich, dass ich die vorliegende mit dem Titel: „Workflows zur binauralen Audioproduktion in Ableton Live: Untersuchung ausgewählter Softwarelösungen“ selbstständig und ohne fremde Hilfe verfasst und keine anderen als die angegebenen Hilfsmittel benutzt habe.

Die Stellen der Arbeit, die dem Wortlaut oder dem Sinn nach anderen Werken entnommen wurden, sind in jedem Fall unter Angabe der Quelle kenntlich gemacht. Ebenso sind alle Stellen, die mit Hilfe eines KI-basierten Schreibwerkzeugs erstellt oder überarbeitet wurden, kenntlich gemacht. Die Arbeit ist noch nicht veröffentlicht oder in anderer Form als Prüfungsleistung vorgelegt worden.

Ich habe die Bedeutung der ehrenwörtlichen Versicherung und die prüfungsrechtlichen Folgen (§ 24 Abs. 2 Bachelor-SPO, § 23 Abs. 2 Master-SPO (Vollzeit)) einer unrichtigen oder unvollständigen ehrenwörtlichen Versicherung zur Kenntnis genommen.

Stuttgart, den 28.05.2025



The image shows a handwritten signature in black ink on a horizontal line. The signature is stylized and appears to read 'T Haas'.