HAW HAMBURG

Bachelorarbeit

Benjamin Oechsle

Entwicklung eines binauralen 3D-Sound-Positionierungs-Algorithmus durch Zuhilfenahme von Head-Related Transfer Functions

Fakultät Technik und Informatik
Department Informatik

Faculty of Computer Science and Engineering
Department Computer Science

Benjamin Oechsle

Entwicklung eines binauralen 3D-Sound-Positionierungs-Algorithmus durch Zuhilfenahme von Head-Related Transfer Functions

Bachelorarbeit eingereicht im Rahmen der Bachelorprüfung
im Studiengang *Bachelor of Science Angewandte Informatik*
am Department Informatik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer: Prof. Dr. Peer Stelldinger
Zweitgutachter: Prof. Dr. Stephan Pareigis

Eingereicht am: 16.08.2022

Benjamin Oechsle

Thema der Arbeit

Entwicklung eines binauralen 3D-Sound-Positionierungs-Algorithmus durch Zuhilfenahme von Head-Related Transfer Functions

Stichworte

Binaural, 3D-Sound, Räumliches Hören, JUCE, Audio, Positionierungs-Algorithmus, Head-Related Transfer Functions, C++, Faltung, FIR-Filter, Fourier Transformation

Kurzzusammenfassung

Binaurale Algorithmen spielen auf dem wachsenden Markt der Augmented- und Virtual-Reality-Anwendungen eine zunehmend wichtige Rolle. Um diese Algorithmen besser einordnen zu können, wird in erster Instanz der auditive Kortex und das räumliche Hören untersucht und bewertet. Dazu wird sich in dieser Arbeit mit der Einführung in die binaurale Ortung und deren Abbildung als Head-Related Transfer Functions (HRTF) in der digitalen Domäne beschäftigt. Im weiteren Verlauf wird die Audiosignalverarbeitung in Computersystemen und der Einsatz von Filtern, insbesondere Finite-Impulse-Response-Filtern (FIR-Filter) betrachtet. Auf Basis der FIR-Filter wird nachgehend ein binauraler Algorithmus entwickelt, der die Komponenten des räumlichen Hörens simulieren und so einen dreidimensionalen virtuellen Raum abbilden kann.

Benjamin Oechsle

Title of Thesis

Development of a binaural 3D-Sound algorithm with Head-Related Transfer Functions

Keywords

Binaural, 3D-Sound, Spatial Hearing, JUCE, Audio, Localization-algorithm, Head-Related Transfer Functions, C++, Convolution, FIR-Filter, Fourier Transform

Abstract

Binaural algorithms are playing an increasingly important role in the growing market of augmented and virtual reality applications. In order to classify these algorithms, the auditory cortex and spatial hearing are being studied and evaluated. For this purpose, this thesis will focus on the introduction to binaural localization and its mapping as Head-Related Transfer Functions (HRTF) in the digital domain. Further on, audio signal processing in computer systems and the use of filters, especially finite impulse response filters (FIR filters) are considered. Based on the FIR filters, an binaural algorithm is subsequently developed, that can simulate the components of spatial hearing in a three-dimensional virtual space.

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

1 Einleitung

Das auditive System zählt neben dem visuellen Kortex zu einem der wichtigsten kognitiven Sinne des Menschen. Dabei ist der Mensch durch das räumliche Hören in der Lage, einen detaillierten akustischen Wahrnehmungsraum abzuleiten, um sich in einer komplexen Umwelt zu orientieren [23].
Das Gehör ist dabei im Stande, eine große Menge verschiedener Informationen simultan zu erfassen und zu prozessieren. Unabhängig davon, ob es sich um soziale Interaktion durch Sprache handelt, bei der das Gehirn in der Lage ist, eine Schallquelle aus vielen verschiedenen Geräuschen zu isolieren (Cocktail-Party Effekt), oder der Orientierung und Einschätzung der eigenen Umwelt dient. Dabei ist es nicht nur in der Lage, Informationen zum Inhalt zu interpretieren, sondern kann über die ausgewerteten Metainformationen Schallereignisorte exakt im Raum lokalisieren und durch möglichst akkurate Positionierung einen eigenen akustischen Wahrnehmungsraum ableiten.

Das räumliche Hören, in der Literatur auch binaurales Hören genannt, ist bereits seit den 1960er Jahren Gegenstand der Forschung und wurde in unterschiedlichen Studien eingehend untersucht. Seit dem Aufkommen von digitalen Computersystemen hat sich der Forschungsbereich um die Simulation räumlichen Hörens erweitert. Ab den 1980er Jahren wurden mit kopfbezogenen Übertragungsfunktionen den sogenannten Head-Related Transfer Functions (im Folgenden HRTF) erste Ansätze formuliert, um Algorithmen zur Positionierung von Schallquellen in einem virtuellen dreidimensionalen Raum zu entwickeln. Dieser in der Wissenschaft viel diskutierte Ansatz ermöglicht es inzwischen, jedes beliebige Geräusch in einem virtuellen Raum zu positionieren und über Kopfhörer nachzuempfinden.

1.1 Motivation

Durch die zunehmende Präsenz von Augmented-Reality (AR) und Virtual-Reality (VR) Anwendungen werden ganz neue Anforderungen an Audiosignale innerhalb dieser Applikationen gestellt. Sei es bei Computerspielen, Videos, Musik oder Mixed-Reality-Applikationen, in VR-Anwendungen muss stets gewährleistet sein, dass ein korrekter dreidimensionaler Höreindruck entsteht, auch wenn der Ton über Kopfhörer abgespielt wird. Dafür zuständig sind die binauralen Algorithmen, die mittlerweile verstärkt in den Fokus von Audioprogrammierern und Anwendungsentwicklern gerückt und in dieser Arbeit untersucht werden sollen.

1.2 Ziele und Umfang

In dieser Arbeit wird anhand von Untersuchungen zur Schalllokalisation und den Möglichkeiten der digitalen Signalverarbeitung ein Algorithmus zur dynamischen Positionierung von Schallereignissen in einem virtuellen dreidimensionalen Raum skizziert. Dazu werden die verschiedenen Faktoren der Schalllokalisation untersucht und deren Auswirkung auf die räumliche Ortung veranschaulicht. Insbesondere wird aufgezeigt, dass neben äußeren Gegebenheiten, wie Reflexion und Abstrahlverhalten von begrenzenden schallharten Oberflächen, die anatomischen Merkmale und damit verbundene individuelle Schallreflexionen eine entscheidende Rolle für die Lokalisierung von Schallereignissen spielen. Es wird die Erfassung und das Persistieren solcher Merkmale in speziellen HRTF-Dateien im Detail betrachtet und deren Verwendung innerhalb einer Applikation zur Bearbeitung von Audiosignalen erarbeitet. Des Weiteren wird auf die Besonderheiten der digitalen Audiosignalverarbeitung und den Einsatz unterschiedlicher Filtertypen eingegangen und deren Einsatz als Außenohrübertragungsfunktion mittels der HRTF-Koeffizienten veranschaulicht. Mit dem entwickelten Algorithmus soll es möglich sein, ein beliebiges Audiosignal in einem dreidimensionalen Raum abzuspielen und die definierte Position nachzuempfinden.

1.3 Struktur der Arbeit

Die Arbeit ist in acht Kapitel aufgeteilt. Nach einer Einleitung wird das auditive System des Menschen und die konzeptionellen Grundlagen des räumlichen Hörens erörtert und auf Studien zu verschiedenen Ansätzen der Lokalisierung eingegangen. Des Weiteren werden die Entfernungswahrnehmung und Leitlinien zur Wahrnehmung von Klängen in reflexiven Umgebungen aufgezeigt.
In diesem Kontext wird die Bedeutung von HRTFs für die räumliche Ortung von Schallereignissen und deren Einsatz in binauralen Audioapplikationen eingeführt. Dabei werden sowohl die Methodik zum Erstellen solcher Funktionen beleuchtet als auch deren zeitliche Abbildung, die Head-Related Impulse Responses (im Folgenden HRIR).
Das darauffolgende Kapitel geht sowohl auf die Wandlung von analogen zu digitalen Audiosignalen ein (A/D Wandlung) als auch auf die digitale Signalverarbeitung (engl. Digital Signal Processing) und deren Bearbeitung durch verschiedene Filterdesigns. Zu diesem Zweck werden die bei Audiofiltern eingesetzte Fourier-Transformation und der Unterschied zur schnellen Fourier-Transformation erläutert.
Im nächsten Schritt wird die Modellierung eines Algorithmus beschrieben, der in der Lage ist, ein Mono-Eingangssignal in Abhängigkeit einer gewählten Position in ein binaurales Stereosignal zu überführen. Im Anschluss wird die Implementierung des Algorithmus skizziert, wobei neben einer detaillierten Betrachtung des Anwendungsdesigns auch auf die Wahl der verwendeten Programmiersprache C++ und das darauf aufsetzende JUCE-Framework eingegangen wird. Weiterführend wird der entwickelte Algorithmus anhand von Tester:innen evaluiert, bevor im letzten Kapitel eine Zusammenfassung und ein Ausblick auf weitere Forschungen zu dem Thema gegeben wird.

2 Räumliches Hören

Dieses Kapitel definiert die Grundlagen des menschlichen Hörens. Dazu wird nach einer Einführung in das menschliche Gehör und die Bedeutung der einzelnen Komponenten des auditiven Kortex auf die verschiedenen Lokalisierungsebenen und deren Charakteristika eingegangen. Im Anschluss werden die Begriffe der binauralen, sowie der monoauralen Schalllokalisation vorgestellt und veranschaulicht, bevor zum Abschluss auf die Entfernungswahrnehmung und Sonderfälle eingegangen wird.

2.1 Das menschliche Gehör

Auditive räumliche Ortung bezeichnet die Fähigkeit, Schallereignisse in einem dreidimensionalen Raum anhand von physikalischen und anatomischen Eigenschaften identifizieren und zuordnen zu können [22]. Um die Komplexität des räumlichen Hörens und damit die Emulation von binauralen Audiosignalen besser verstehen zu können, muss die essentielle Funktionsweise des menschlichen Gehörs bewusst gemacht werden. Abbildung 2.1 zeigt die diagrammatische Darstellung eines menschlichen Ohrs im Querschnitt.
Das auditive System ist ein äußerst komplexes Sinnesorgan, welches in die drei grundlegenden Abschnitte Außenohr, Mittelohr und Innenohr unterteilt werden kann. Schall, der als Longitudinalwelle (Welle, deren Schwingungen parallel zur Ausbreitungsrichtung verlaufen [27]) auf das Außenohr trifft, wird durch die Ohrmuschel (engl. Pinna), welche den Eingang des Gehörkanals umschließt, in den äußeren Gehörgang weitergeleitet. Der Gehörgang, ein im Durchschnitt 25mm langer, leicht gebogener und mit Haut bespannter Kanal, leitet den Schall über Reflexionen bis in das Mittelohr weiter. Dort treffen die Schallwellen dann auf eine dünne, meist runde bis leicht ovale Hautmembran (Trommelfell), die in Schwingung versetzt wird und den äußeren Gehörgang vom Mittelohr trennt. Das Trommelfell gibt die empfangenen Schwingungen anschließend an die Gehörknöchelchen Hammer, Amboss und Steigbügel weiter, die den Schall um das zwanzig- bis fünfzigfache verstärken. Anschließend werden die verstärkten Schwingungen an das

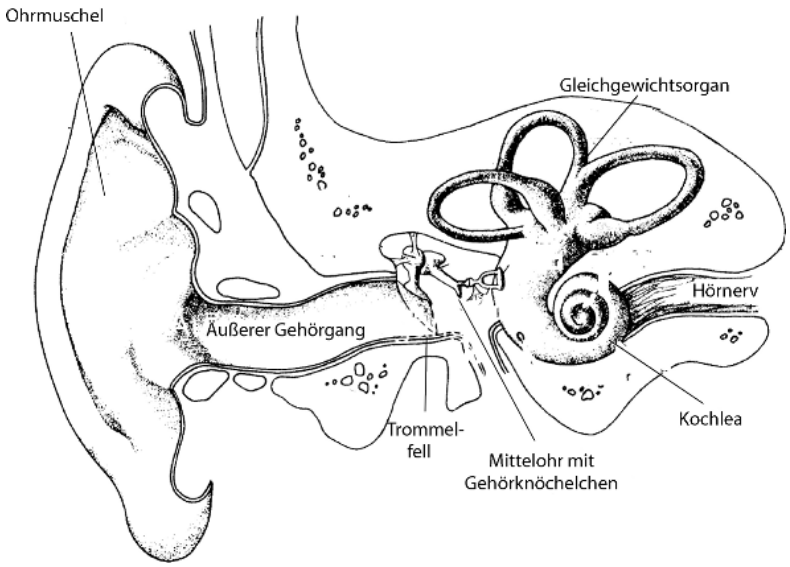

Abbildung 2.1: Das menschliche Ohr mit den zur Verarbeitung von Schall relevanten Bereichen [27]

Innenohr übertragen, wo sie in der Schnecke (engl. Cochlea) in elektrische Impulse gewandelt, und über den Hörnerv an das Hörzentrum des Gehirns weitergeleitet werden. Das Hörzentrum verarbeitet die räumlichen Informationen zum Schallereignisort und interpretiert daraus die Richtung und Entfernung zum Ohr [14].
Für die Entwicklung eines binauralen 3D-Sound-Algorithmus ist vor allem der Bereich des Außenohrs bis hin zum Trommelfell relevant und wird daher fokussiert betrachtet. Speziell wird die individuelle Formung der Ohrmuschel, als auch die Auswirkungen von Schallüberlagerungen und daraus resultierenden Kammfiltereffekten im äußeren Gehörgang durch den Kopf und Oberkörper untersucht und bewertet. Die hier gewonnen Erkenntnisse sind elementar für die Gewinnung und das Verständnis der HRTFs, die bei der korrekten Ortung von Schallereignissen und somit der Implementierung des Algorithmus eine entscheidende Rolle spielen.
Die Ohrmuschel, die das offene Ende des äußeren Gehörkanals und damit den Einstiegspunkt in das menschliche Hörorgan bildet, befindet sich in der Regel auf identischer Höhe, jeweils links und rechts, an der Seite des Kopfes und kann als eine Art Antenne mit richtungs-, frequenz- und distanzabhängiger Richtcharakteristik für den eintreffenden Schall verstanden werden [27]. Dabei dient sie mit einem Winkel zwischen 25° und 45° nicht nur, wie lange vermutet [14], als eine Art Trichter, der den eintreffenden Schall bündelt und in den Gehörgang weiterleitet, sondern auch als ein linearer Filter, dessen Übertragungsfunktion die ankommenden Schallwellen in Abhängigkeit zum relativen

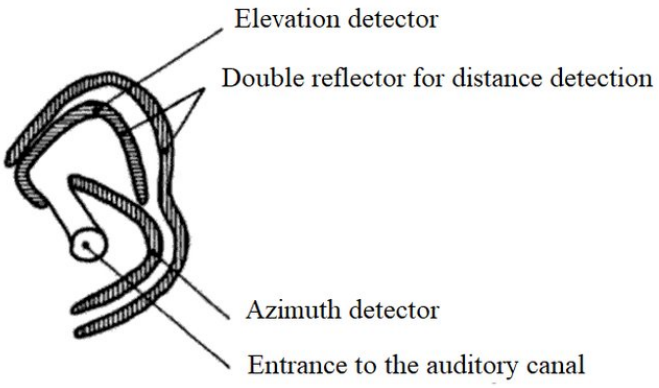

Abbildung 2.2: Eine analytische Darstellung der Ohrmuschel und der Funktion der einzelnen Bereiche [14]

Ursprung und der Entfernung des Hörers stark beeinflusst. Die individuell geformte Ohrmuschel kodiert dabei die eintreffenden Schallereignisse in bestimmte Interferenzmuster, die durch das Gehirn anschließend mit erlernten und abgespeicherten Mustern verglichen werden. Diese Interferenzmuster sind entscheidend für eine räumliche Einordnung auf der vertikalen Achse in der Medianebene (siehe Kapitel 2.2). Die Überlagerung von Direktschall und Reflexionen und die daraus resultierenden Interferenzmuster spielen bei der Entwicklung von binauralen Algorithmen eine signifikante Rolle und können als sogenannte Head-Related Transfer Functions (HRTF) (Näheres dazu in Kapitel 3) mathematisch beschrieben werden.
Der bis in die 1960er Jahre verbreiteten Annahme, dass die Ohrmuschel in Relation zu den eintreffenden Schallwellen zu klein und damit unbedeutend für Reflexionen und die Bildung von Schallschatten ist, geht Batteau in der Arbeit von 1967 The Role of the Pinna in Human Localization nach [11]. Batteau erkennt, dass die Interferenzmuster, die durch Direktschall und den von der Ohrmuschel selektiv reflektierten Schall, in Abhängigkeit zur Entfernung und Position der Schallquelle stehen und somit essenziell für die Bildung eines präzisen Hörereignisortes sind. Dabei werden in den Strukturen der Ohrmuschel relativ zur Schalleinfallsrichtung unterschiedliche Resonanzen angeregt und das so entstehende einzigartige Resonanzmuster im Gehirn ausgewertet. Abbildung 2.2 zeigt die von Batteau identifizierten Bereiche der Ohrmuschel und deren Effekt auf die räumliche Abbildung des Schallereignisses.

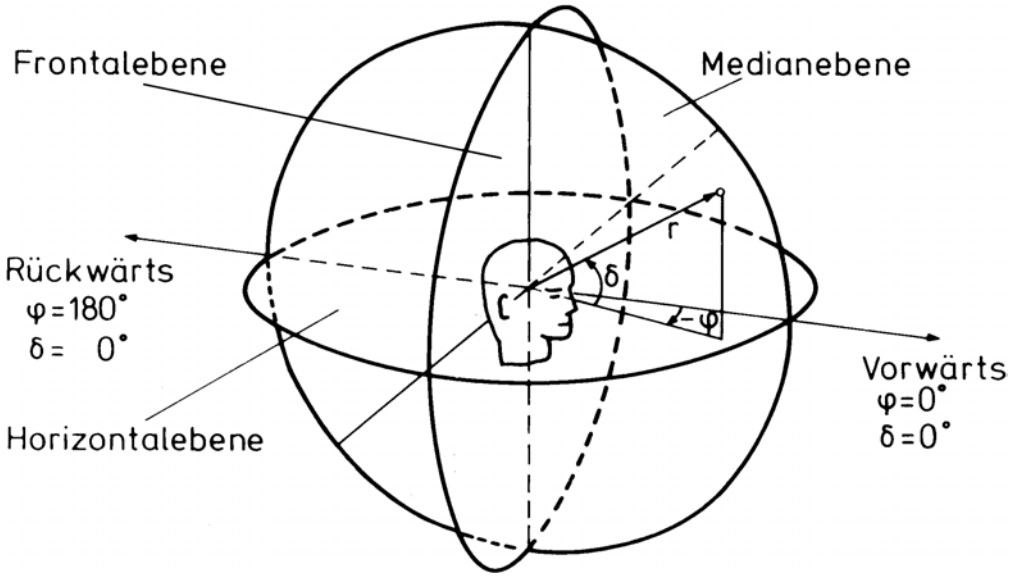

Abbildung 2.3: Kopfbezogenes Koordinatensystem mit den Lokalisationsebenen [27]

2.2 Lokalisationsebenen

In diesem Abschnitt wird detailliert auf die verschiedenen Ebenen der Schalllokalisation eingegangen und deren Präzision verglichen. Weiterhin werden wesentliche Begriffe der Schalllokalisation eingeführt. Die Schalllokalisation beschreibt nach Jens Blauert die Korrelation zwischen einem Schallereignis und dem individuellen Hörereignis, bei welchem die tatsächliche Position der wahrgenommenen Position entspricht [14]. Lokalisierungsunschärfe wird nach Blauert als die Abweichung der Position der physischen Schallquelle vom wahrgenommenen Hörereignisort beschrieben.
Um die Positionierung von Schallereignissen in einem dreidimensionalen Raum systematisch beschreiben zu können, wird deren Ursprung häufig durch einen dreistelligen Vektor beschrieben, der die Entfernung r, sowie Azimut φ und Elevation δ beinhaltet. Für die Abbildung des Schallereignisses im Raum bedient man sich dabei eines, wie in Abbildung 2.3 abgebildeten, kopfbezogenen Polarkoordinatensystems, dessen Ursprung sich auf der interauralen Achse zwischen den beiden Gehörkanaleingängen befindet [15].

2.2.1 Die Horizontalebene

Die Horizontalebene oder auch Azimutebene teilt den Kopf dabei in einen oberen und einen unteren Bereich, so dass die Schalleinfallsrichtung von vorne, rechts, hinten oder

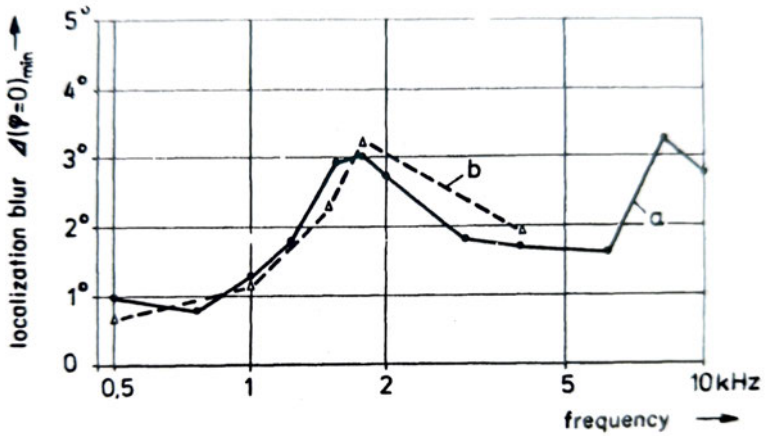

Abbildung 2.4: Lokalisierungsunschärfe bei 0°φ über den Frequenzbereich von 500Hz bis 10KHz. a: Sinuston mit 50db Schalldruck. b: Gaußverteilter Tonimpuls 1/3 Oktave [14]

links zugeordnet werden kann. Wenn ein Schallereignis direkt von vorne auf den Hörenden trifft, entspricht das einem Einfallswinkel von 0°Azimut (φ). Der Azimutwinkel steigt dann im Uhrzeigersinn, bis er bei 180°φ ein direkt von hinten wahrgenommenes Schallereignis beschreibt. Auf der Horizontalebene ist der Mensch in der Lage die größte Lokalisationsschärfe aufzuweisen, so dass er im Stande ist, Abweichungen von einem Grad bei direkt von vorne eintreffenden Schallereignissen wahrzunehmen [19]. Die Lokalisierungsunschärfe hängt von der Frequenz des Schallereignisses ab und bewegt sich zwischen unter einem bis hin zu drei Grad, wie Abbildung 2.4 zeigt.

2.2.2 Die Frontalebene

Die Frontalebene oder auch Coronalebene teilt entlang der interauralen Achse den Kopf in einen vorderen und einen hinteren Bereich und liegt orthogonal zur Horizontalebene. Die Schalleinfallsrichtung kann also von links oder rechts, sowie von oben oder unten erfolgen. Die Bewegung auf der Ordinate wird dabei in Grad Elevation δ angegeben [27].

2.2.3 Die Medianebene

Die Medianebene oder auch Sagittalebene liegt orthogonal zu den beiden anderen Ebenen und teilt den Kopf in eine linke und eine rechte Hälfte. Einfallsrichtungen für den Schall sind hier vorne, oben, hinten und unten. Da diese Ebene sich auf 0°φ befindet und den

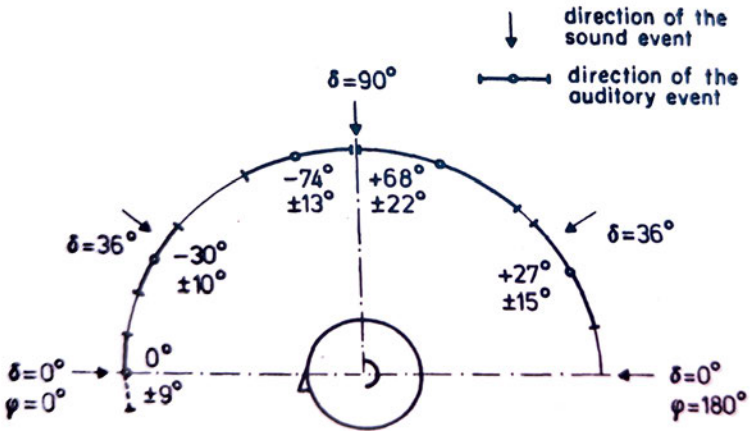

Abbildung 2.5: Lokalisierungsunschärfe in der Medianebene bei kontinuierlicher Sprache einer bekannten Person [14]

Ursprung schneidet, erfolgt die Lokalisation auf der Medianebene grundlegend anders als auf der Horizontalebene. Hier gibt es auf Grund der zentralen Position keinen zeitlichen Versatz zwischen der Ankunft eines Schallereignisses zwischen den beiden Ohren. So ist die Bestimmung des δ-Winkels einzig durch die einmaligen Interferenzmuster, die durch Reflexions- und Absorptionseffekte des Körpers, des Kopfes und der Ohrmuschel erzeugt werden, möglich. In Abbildung 2.5 lässt sich erkennen, dass die Lokalisierungsunschärfe in der Medianebene sich zwischen 9° bei 0°δ bis hin zu 22° bei 90°δ bewegt. Damit ist die Lokalisierungsgenauigkeit in der Medianebene deutlich undifferenzierter als in der Horizontalebene [14].

2.3 Auditive Ortung

Die Ortung von Schallereignissen wird in zwei grundlegende Kategorien unterteilt. Die auditive Ortung durch unterschiedliche Wahrnehmung auf beiden Ohren wird als interaurale oder binaurale Lokalisation definiert. Bei einer Ankunft identischer Signale an beiden Ohren wird die Ortung hingegen als monoaurale Lokalisation beschrieben [14]. Dabei gilt Ersteres als das dominante Lokalisierungsmerkmal zur Ortung von Schallereignissen im menschlichen Gehör. Zu Letzterem zählen Schallereignisse, die auf der Medianebene liegen. Sie lassen sich durch die Symmetrie des Kopfes nicht durch Laufzeit oder Pegelunterschiede differenzieren und werden ausschließlich über den Unterschied von Interferenzmustern interpretiert. Bei Schallereignissen, die nicht auf 0°φ oder 180°φ lie-

gen, spricht man von interauralen Schallereignissen. So spielen die folgende Pegeldifferenz und Laufzeitdifferenz bei diesen Signalen eine entscheidende Rolle.

2.3.1 Pegeldifferenz

Die interaurale Pegeldifferenz oder auch Schalldruckdifferenz (engl. Interaural Level Difference ILD, in der Literatur auch Interaural Intensity Difference IID genannt) ΔL beschreibt die Dämpfung des Schallereignisses zwischen dem der Schallquelle zugewandten und der Schallquelle abgekehrten Ohr. Sie gilt als die älteste Theorie für räumliches Hören und wurde schon von Rayleigh und Steinhauser im Jahre 1877 erwähnt [14]. Beim Auftreten der interauralen Pegeldifferenz fungiert der Kopf bei hohen Frequenzen als ein Schallhindernis, das auf dem Ohr der zugewandten Seite einen Druckstau und auf dem gegenüberliegenden einen Schallschatten entstehen lässt. Dieser Effekt lässt sich am besten bei hohen Frequenzen beobachten, deren Wellenlänge unterhalb des Kopfdurchmessers liegt. Bei der Annahme, dass ein durchschnittlicher Kopf einen Durchmesser von etwas weniger als 20 cm aufweist, sollte das Schallereignis eine Grenzfrequenz von 1600 Hz (Wellenlänge λ 21.4 cm) aufweisen oder ein Obertonspektrum überhalb dieser Frequenz beinhalten, um eine wirkungsvolle ILD zu erzeugen und eine fehlerfreie Ortung zu gewährleisten [19]. Bei höheren Frequenzen kann der Schalldruckunterschied zwischen den Ohren von 10 db bei 3 kHz auf bis zu 35 db bei 10 kHz ansteigen [22].

2.3.2 Laufzeitdifferenz

Die interaurale Laufzeitdifferenz (engl. Interaural Time Difference ITD, in der Literatur auch Interaural Phase Difference IPD genannt) $\Delta\tau$ tritt dagegen hauptsächlich bei Frequenzen unterhalb von 1600 Hz auf und wird in Abbildung 2.6 dargestellt. Bei diesen Frequenzen ist die Wellenlänge im Verhältnis zum Kopfdurchmesser ausreichend groß, damit sie um den Kopf herum gebeugt werden kann. Dabei wird das Schallereignis bei dem ihm zugewandten Ohr durch die kürzere Laufzeit früher registriert als beim gegenüberliegenden Ohr. Dieser minimale zeitliche Versatz kann vom Hörzentrum im Gehirn ausgewertet werden und trägt maßgeblich zur Ortung auf der Horizontalebene bei. Laut Blauert 1974 beträgt der Umweg, welchen ein Schallereignis mit einem Einfallswinkel von 90°φ zurücklegt, 21 cm. Nimmt man an, dass die Schallgeschwindigkeit c bei 343 m/s liegt, resultiert daraus folgende Laufzeitdifferenz [14]:

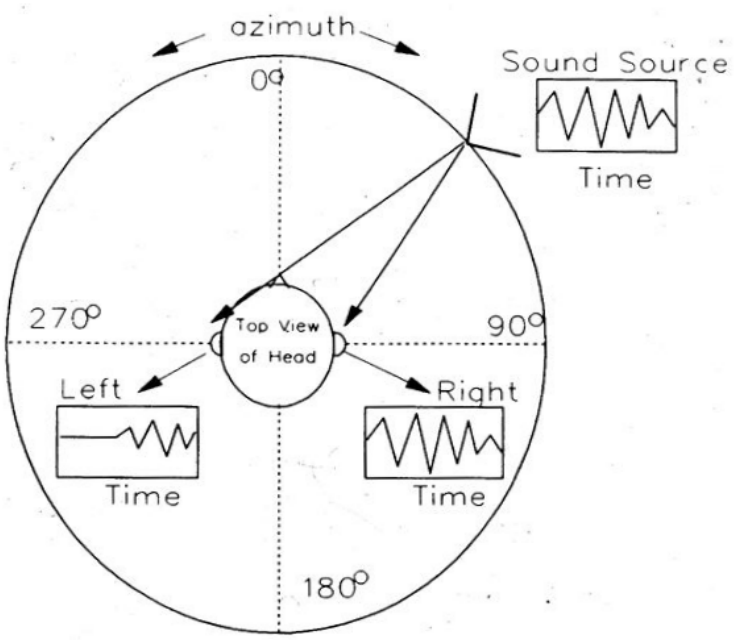

Abbildung 2.6: Laufzeitdifferenz eines Schallereignisses, welches nicht auf der Medianebene liegt und unterhalb von 1600Hz stattfindet [1].

$$\tau = \frac{0,21}{c} = 0,61\mu s$$

2.3.3 Cone of Confusion

Auch wenn die binauralen Lokalisierungsmethoden der interauralen Pegeldifferenz (ILD) und der interauralen Laufzeitdifferenz (ITD) eine große Rolle bei der Ortung des menschlichen Gehörs spielen, kann es bei der vertikalen Lokalisation und der Differenzierung von Signalen, die von vorne und hinten kommen, zu Mehrdeutigkeiten führen. Schallereignisse, die sich auf der Oberfläche eines imaginären Kegels befinden, dem sogenannten Cone of Confusion, werden wie in Abbildung 2.7 gezeigt, häufig falsch gedeutet. Dies liegt an den fehlenden binauralen Merkmalen durch identische Laufzeit und identischen Pegel, den das Schallereignis und dessen Spiegelereignis im Kegel aufweisen [22].

Dabei wird davon ausgegangen, dass der Kopf eine perfekte Sphäre ist. Auch der Ortungsmechanismus durch die monoaurale Signalerkennung (eingeführt in Abschnitt 2.4) kann das Problem des Cone of Confusion nicht vollständig adressieren. Studien haben

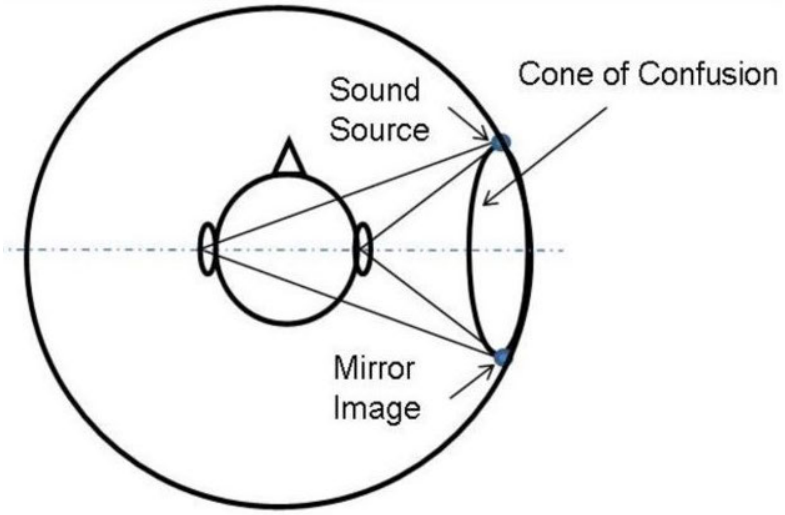

Abbildung 2.7: Der Cone of Confusion entsteht, wenn Schallereignisse durch symmetrische Anordnung entlang der Ohrachse gleiche ILD und ITD aufweisen [22].

ergeben, dass der Mensch das Problem durch eine leichte Kopfbewegung löst, die auf einer der betroffenen Achsen ausgeführt wird, die in Abbildung 2.8 veranschaulicht wird. Diese Kopfbewegung wird dabei unterbewusst ausgeführt und dient dazu die ILD und ITD des betroffenen Schallereignisses zu vergrößern, um eine bessere Ortung zu generieren. Um diese Änderung der Kopfposition und das erneute korrekte Verarbeiten des Schallereignisses zu gewährleisten, sollte die Dauer des Schallereignisses mehr als 200 ms betragen.

(a) Horizontale Bewegung (b) Vertikale Neigung des Kopfes

Abbildung 2.8: Bewegungen des Kopfes, um die ILD und ITD zur Lokalisierung im Cone of Confusion nutzen zu können [5].

2.4 Monoaurale Lokalisierung

Die Lokalisierung in der Medianebene stellt einen Sonderfall dar, da durch den gleichen Abstand des Schallereignisses zu beiden Ohren und die Symmetrie des Kopfes weder eine Pegeldifferenz (ILD) noch eine Laufzeitdifferenz (ITD) zur Lokalisierung herangezogen werden kann.
Forschungsergebnisse von Carsten und Salinger 1922 haben ergeben, dass es durch die fehlenden interauralen Unterschiede bei Schallereignissen in der Medianebene häufig zu Abweichungen zwischen Schall- zu Hörereignisort kommt [14]. Weitere Forschungen durch Blauert 1974 haben ergeben, dass die Abweichungen in der Medianebene stark von der Art der Schallquelle abhängen. So werden breitbandige Schallereignisse häufiger richtig zugeordnet als schmalbandige. Auch die Dauer und Wiederholung, respektive Bekanntheit, des Audiosignals trägt zur korrekten Ortung bei. Roffler und Buttler entdeckten 1968 bei einem Versuch, dass ein Schallereignis, welches auf der Horizontalebene bei $0°\varphi$ angeordnet ist und einen Sinus-Sweep (ein Sinussignal, dessen Frequenz sich exponentiell über die Zeit steigert) von 200Hz bis 16kHz vollzieht, in der Wahrnehmung mehrere Male die Position von vorne nach hinten wechselt.
Bei weiterführenden Experimenten, bei denen Versuchspersonen mit schmalbandigen Terzfrequenzen aus den Richtungen vorne, hinten und oben beschallt wurden und anschließend das Hörereignis auf der Medianebene bestimmt werden sollte, kam Blauert zu dem Schluss, dass die Hörereignisposition nicht von der Schalleinfallsrichtung, sondern von der Terzmittenfrequenz abhängt. Die daraus resultierenden richtungsbestimmenden Bänder, auch Blauert'sche Bänder genannt, haben bis heute eine große Bedeutung bei der Bearbeitung von Audiosignalen und in der Tontechnik. Frequenzen im Bereich 250Hz bis 500Hz und um 3,5kHz werden dabei als vor dem Kopf lokalisiert. Bei Frequenzen um 1kHz und 12kHz wird das Schallereigniss stattdessen von hinten und bei 8kHz von oben wahrgenommen wie in Abbildung 2.9 veranschaulicht wird. Da bei breitbandigen Signalen eine bessere Ortbarkeit auf der Medianebene vorliegt, geht Blauert davon aus, dass die Übertragungsfunktion des Kopfes und der Ohrmuschel auf entsprechend einfallenden Schall eine ähnliche Filterwirkung aufweist wie die Blauert'schen Bänder [27].

2.5 Entfernungswahrnehmung

Die Entfernungswahrnehmung ist der Versuch des Gehirns Schallereignisse durch den Abgleich mit abgespeicherten Interferenzmustern in der Entfernung zum Hörenden zu

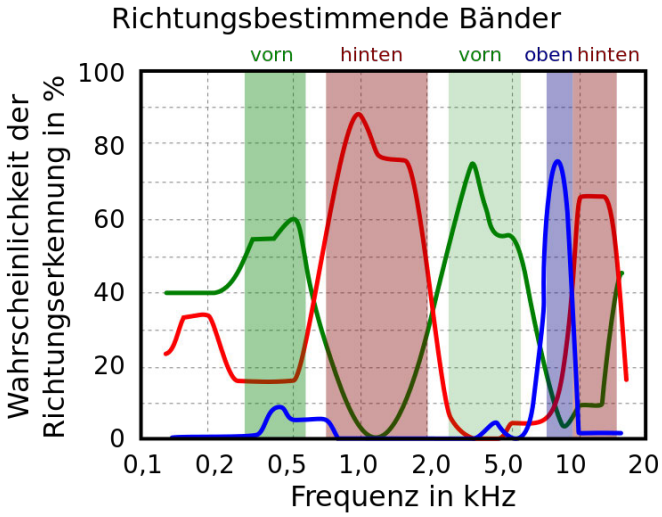

Abbildung 2.9: Blauert'sche Bänder, welche die richtungsbestimmenden Frequenzbänder angeben [10].

positionieren. Dabei muss zwischen nicht-reflektierenden und reflektierenden als auch zwischen Schallereignissen im Nahfeld (unter 1 m) und solchen mit einer Entfernung von über einem Meter zum Hörer unterschieden werden. Im Folgenden wird die Entfernungswahrnehmung in einer nicht-reflektierenden Umgebung betrachtet. Im Nahbereich ändern sich bei Veränderung der Entfernung die interauralen und monoauralen Merkmale des Hörereignisses massiv, da in diesem Bereich der Kopf eine schirmende Wirkung besitzt und der relative Abstand vom Schallereignis zum zugewandten Ohr deutlich kleiner ist als der zur abgewandten Seite. Bei größerer Entfernung tritt in einer Freifeldsituation neben einer Abnahme des Schalldruckpegels von 6 db pro verdoppelter Entfernung eine zusätzliche Absorption hoher Frequenzen auf, so dass Schallereignisse dumpf wirken [23]. Formel 2.1 beschreibt die logarithmische Abnahme des Schalldruckpegels ΔSP bei einer Verdopplung der Entfernung von D_1 zu D_2 um 6 db.

$$\Delta SP = 20 * log\frac{D_1}{D_2} \tag{2.1}$$

In reflektierenden Umgebungen, wie sie im Alltag zumeist anzutreffen sind, spielt die Schallreflexion eine entscheidende Rolle für die Entfernungswahrnehmung. Die Direct-to-Reverberant-Ratio (DDR) beschreibt die Relation des Schalldruckpegels zwischen dem direkt von der Schallquelle emittierten Signal und den Reflexionen. Diese Lautstärkeunterschiede werden dabei vom Gehirn ausgewertet und zur Entfernungslokalisation her-

angezogen. Im Rahmen der bisherigen Betrachtungen wurde davon ausgegangen, dass eine einzelne Schallquelle vorliegt. Da bei Reflexionen jedoch die Annahme getroffen werden muss, dass jede Reflexion als eine eigene Schallquelle zu betrachten ist, stellt sich die Frage, weshalb in reflektierenden Räumen trotzdem von einer fehlerfreien Ortung des Schallereignisses ausgegangen werden kann.
Um auch in einem reflektierten Raum mit schallharten Grenzflächen ein Schallereignis korrekt orten zu können, nutzt das Gehirn den Präzedenzeffekt (Henry 1849, Wallach et al. 1949) (engl. Precendence Effect). Auch Gesetz der ersten Wellenfront genannt. Dieser besagt, dass wenn ein gleiches Schallereignis in kurzem Abstand ($<$ 50ms bei Sprache) erneut auf das Gehör trifft, ausschließlich das zuerst eintreffende Signal für die Richtungs- und Entfernungslokalisation herangezogen wird.
Neben den genannten Varianten spielen auch weitere Effekte wie die visuelle Komponente, dynamische Lokalisierung durch Kopfbewegungen und Erinnerungseffekte eine wichtige Rolle in der Schalllokalisation. Da die Bearbeitung dieser Effekte den Umfang der Arbeit übersteigen würde, werden hier aber ausschließlich die oben genannten Effekte behandelt.

In diesem Kapitel wurden die Grundlagen für eine erfolgreiche Positionierung von Schallereignissen im dreidimensionalen Raum gelegt, indem die Funktionsweise des auditiven Kortex und die unterschiedlichen Lokalisierungsmethoden erarbeitet und veranschaulicht wurden. Dabei wurden sowohl die binaurale Lokalisation und deren Merkmale wie ILD und ITD, als auch monoaurale Lokalisationsmethoden der Medianebene vorgestellt.

3 Head-Related Transfer Functions

Die im vorhergehenden Kapitel 2 beschriebenen Lokalisierungsvarianten sind auf Grund der einzigartigen anatomischen Eigenschaften von Ohrmuschel, Kopf und Körperform für jeden Menschen unterschiedlich ausgeprägt und werden in den Head-Related Transfer Functions abgebildet. In diesem Kapitel sollen die HRTF und deren zeitliche Repräsentation die HRIR vorgestellt und ein Einblick in die Erstellung von HRTF-Datenbanken gegeben werden.

3.1 Frequenzdomäne

HRTF sind individuelle, frequenzabhängige Übertragungsfunktionen, die beschreiben, in welcher Art der eintreffende Schall auf dem Weg vom Schallereignis über die Ohrmuschel durch den Gehörgang bis zum Trommelfell reflektiert, absorbiert und gebeugt wird. Dabei wird die spektrale Zusammensetzung des Schallereignisses auf Grund von Interferenz- und Kammfiltereffekten bearbeitet. Wie in Abbildung 3.1 zu sehen ist, wirken sich Änderungen in Bezug auf die beiden Ohren in der Horizontalebene deutlich stärker auf den Frequenzgang aus, als Änderungen des Elevation-Winkels in der Medianebene.

3.2 Zeitdomäne

Head-Related Impulse Responses (im folgenden HRIR) stellen die zeitliche Repräsentation der Head-Related Transfer Functions dar und beinhalten die binauralen Faktoren des räumlichen Hörens, wie ILD und ITD. Durch die Anwendung der inversen Faltung der Frequenzbänder mit dem Originalsignal und der anschließenden inversen Fourier-Transformation (In Kapitel 4.3 beschrieben) wird aus der HRTF die HRIR gebildet. Die so generierte Impulsantwort kann dann in den in dieser Arbeit eingesetzten FIR-Filtern

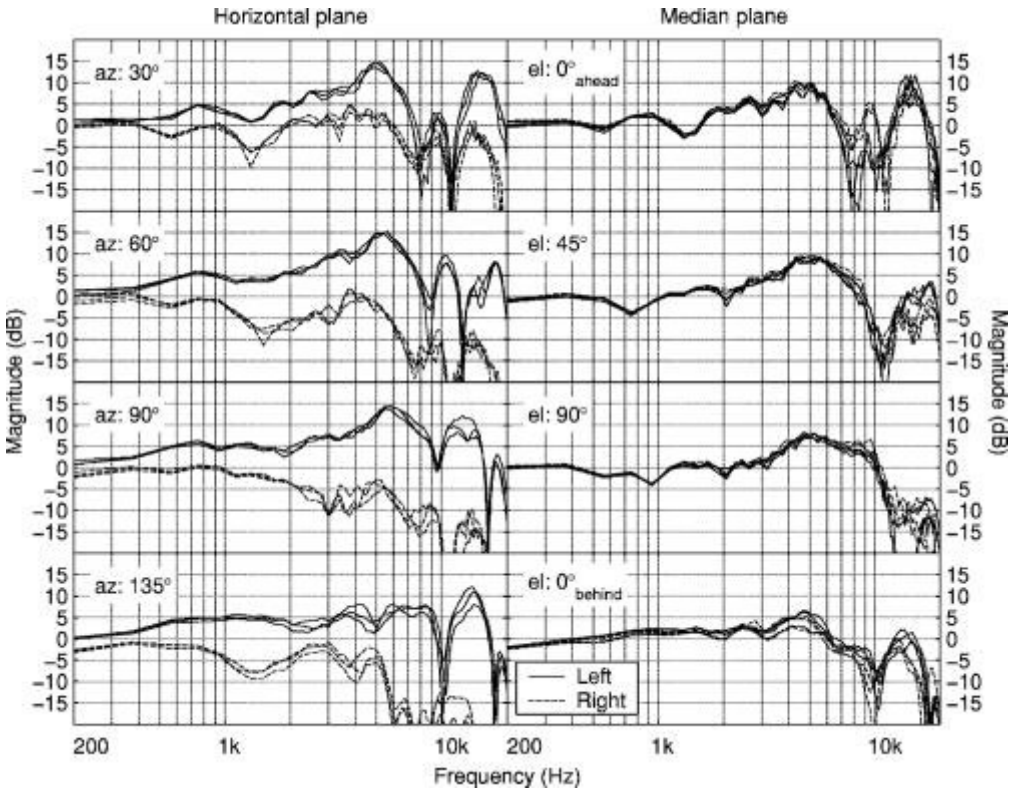

Abbildung 3.1: HRTF-Messungen aus verschiedenen Winkeln. Links in der Horizontalebene. Rechts in der Medianebene [26].

(In Kapitel 4.5.2 beschrieben) als Koeffizientensatz genutzt werden, um ein räumliches Abbild zu generieren.

Zusammen mit der Dämpfung und Verstärkung einzelner Frequenzbänder durch die HRTF beinhaltet das bearbeitete Signal alle räumlichen Informationen eines Schallereignisses und kann vom auditorischen Kortex im Gehirn verarbeitet, mit abgespeicherten Frequenzbildern verglichen und zu einem korrekten Hörereignis ausgewertet werden. Dadurch ist es dem Hörenden möglich, Entfernung, Azimut und Elevation des Schallereignisses korrekt zu bestimmen [23].

3.3 HRTF-Messungen

Auf Grund der Komplexität und der Menge an Faktoren in der Außenohrübertragungsfunktion, ist es bisher nicht gelungen eine exakte mathematische Lösungsfunktion zu ermitteln. Aus diesem Grund wird das Erstellen der HRTF durch experimentelle Messungen in spezialisierten Laborbedingungen realisiert [23]. Dazu werden in einer reflexionsfreien Umgebung, meist ein speziell für diesen Anwendungszweck konstruierter schalltoter Raum, entweder hochsensible, entzerrte Druckempfängermikrofone in den Gehörkanal von Proband:innen eingesetzt oder ein Kunstkopf genutzt, der an der Stelle der Trommelfelle eben solche Mikrofone besitzt. Anschließend werden mit einem Stop- and Go-Verfahren Schallereignisse aus festgelegten Azimut- und Elevation-Winkeln abgespielt und durch die Mikrofone mit den entsprechend ausgeprägten Interferenzmustern aufgenommen.
Ein solch spezialisiertes Messsystem der TU Berlin ist in Abbildung 3.2 abgebildet. In dem vollsphärische Multikanalmesssystem können individuelle HRTF-Messungen durchgeführt und optimiert werden. Dafür wird die Proband:in in der Mitte des 3,50 Meter im Durchmesser betragenden Gerüsts platziert. Nach Anbringung von Mikrofonen in den Gehörgängen können über die Lautsprecher aus verschiedenen δ-Winkeln Schallereignisse abgespielt werden. Zusätzlich kann die Proband:in automatisch in festen Abständen gedreht werden, um verschiedene φ-Winkel zu erhalten. Wie an der vorhergehend beschriebenen Prozedur zu erkennen ist, ist die Erstellung von HRIR-Daten ein zeitaufwändiger Prozess. Dieser Prozess benötigt erfahrenes Personal und kann nur unter den o.g. Laborbedingungen durchgeführt werden.
Da die Durchführung dieser Messungen nicht Teil dieser Ausarbeitung sind, werden die

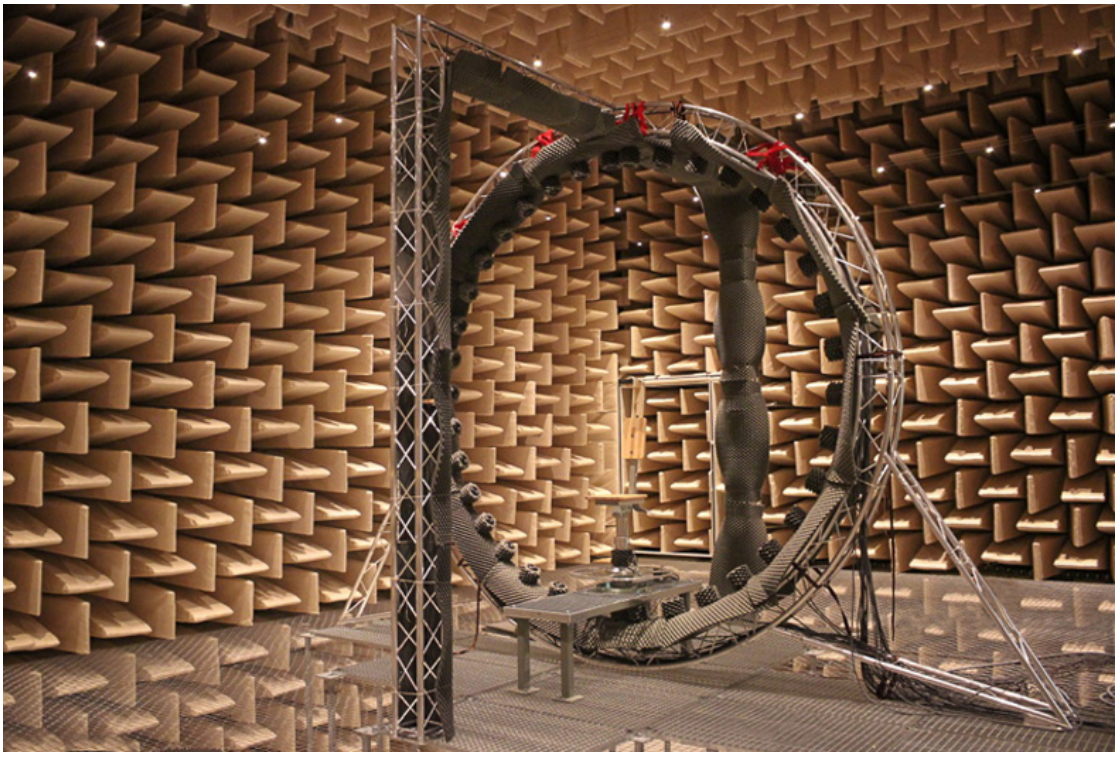

Abbildung 3.2: Vollsphärischer HRTF-Messplatz in der TU Berlin zur Erstellung von HRTF Daten [2].

Berechnungen auf Basis von frei zugänglichen HRTF-Datenbanken durchgeführt. Bis heute existiert dabei noch keine Datenbank, die als Standard für die Entwicklung von binauralen Anwendungen bezeichnet werden kann. Um die im Rahmen dieser Arbeit entstandene Anwendung weiter zu optimieren, würde die Möglichkeit bestehen, für jede einzelne Anwender:in eine individuelle HRTF bereitzustellen [23].

3.4 HRTF-Datenbanken

Wie in dem Kapitel 3.3 beschrieben, existiert eine Reihe an unterschiedlichen Datenbanken für HRTF-Messungen. Um einen besseren Überblick zu erhalten werden in Tabelle 3.1 einige, der nach Blauert aufgelisteten HRTF-Datenbanken verglichen [23].

Nachdem Studien bestätigt haben, dass die HRTF für jeden Menschen individuell sind, wurden weitreichende Messungen und Untersuchungen durchgeführt, um eine möglichst

Name	Institut	Positionen	Azimut	Elevation	Abstand	Auflösung
KEMAR	MIT-Media-Lab	710	5°	-40°+90°	1.4 m	44.100Hz
AUDIS	European Union	2440	15°	-10°+90°	2.4 m	44.100Hz
CIPIC	CIPIC Interface Laboratory	2500	5°	5°	1m	44.100Hz
LISTEN	IRCAM	187	15°	-45°+90°	–	44.100Hz
ARI	Acoustic Research Institute	1550	2.5°	-30°+80°	–	44.100Hz

Tabelle 3.1: Verschiedene HRTF-Datenbanken in der Übersicht

hohe Generalisierung der Übertragungsfunktionen zu erreichen. Ergebnisse dieser Untersuchungen legen nahe, dass es möglich ist, HRTF unter Beschneidung der Fourier-Serie-Repräsentation, auf die ersten 16 Fourier-Komponenten, zu glätten, ohne dabei die Ortbarkeit des Hörereignisses bei den Proband:innen zu beeinflussen. Bei einer weiteren Glättung der spektralen Kurve wurde dagegen eine erhöhte Fehlinterpretation des Elevation-Parameters festgestellt.
Die Forschenden gehen davon aus, dass dies auf eine natürliche Glättung des Frequenzspektrums durch die Ohrmuschel bei hohen δ-Winkeln zurückzuführen ist [23]. Weitere Studien von B. Xie et al. aus dem Jahr 2010 haben gezeigt, dass der Bereich über 8kHz einen essenziellen Einfluss auf die Wahrnehmung und die korrekte Ortung von Schallereignissen nimmt [29].

4 Digitale Signalverarbeitung

Die Digitale Signalverarbeitung (DSV) beschreibt die Analyse und Verarbeitung von digitalisierten Analogwerten und wird in Computersystemen durch eine kontinuierliche Abfolge von Zahlen repräsentiert. Anders als analoge Signalprozessoren kann der Digitalrechner kein zeitkontinuierliches Signal verarbeiten, sondern muss die als analoge Audiowelle vorliegende elektrische Spannung im ersten Schritt durch einen A/D-Konverter (Analog-Digital-Konverter) in ein zeitdiskretes Signal umwandeln. Zu diesem Zweck tastet der Konverter das analoge Signal mit einer Abtastrate (Samplingrate) auf der horizontalen Zeitachse ab und quantisiert die abgetasteten Werte auf der vertikalen Amplitudenachse in ein digitales Signal [30].
Wie in Abbildung 4.1 zu erkennen ist, wird das analoge zeitkontinuierliche Signal $x(t)$ von dem A/D-Konverter in ein zeitdiskretes Signal $x(n)$ konvertiert. Die einzelnen Samplingpunkte werden durch die Striche mit den darauf befindlichen Punkten repräsentiert. Im folgenden Schritt wird $x(n)$ dann von einem Algorithmus um den Faktor 0.5 (6db) in der Amplitude halbiert $y(n)$, bevor es von einem D/A-Konverter wieder in ein zeitkontinuierliches analoges Ausgangssignal $y(t)$ gewandelt wird.

4.1 Abtastfrequenz

Die Abtastfrequenz (Samplerate) beschreibt die Abtastung oder die Auflösung des Rasters der horizontalen Zeitachse, mit der das kontinuierliche, analoge Eingangssignal in ein diskretes Digitalsignal gewandelt wird [30].
Bei der Wahl der Abtastrate muss das Nyquist-Shannon-Abtasttheorem eingehalten werden, welches besagt, dass ein zeitkontinuierliches analoges Signal nur in zeitdiskrete Datenpunkte gewandelt und aus diesen wieder fehlerfrei rekonstruiert werden kann, wenn die Abtastrate $f(s)$ mindestens der doppelten höchsten zu wandelnden Audiofrequenz $f(max)$ entspricht $f(s) > 2 * f(max)$. Um sicherzustellen, dass keine Frequenzen über

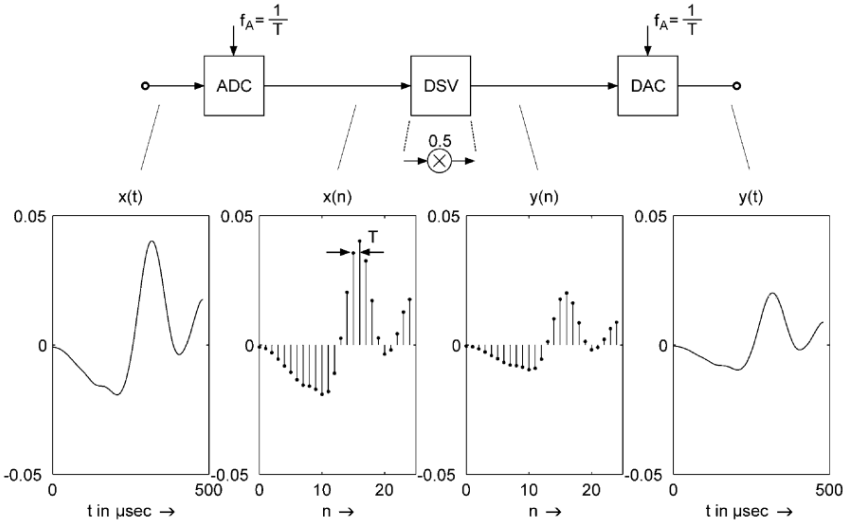

Abbildung 4.1: Wandlung und Bearbeitung eines Audiosignals [30].

$f(max)$ vom A/D-Wandler verarbeitet werden und dadurch eine Spiegelung der Frequenzen $> f(max)$ in den hörbaren Nutzbereich stattfindet, werden Frequenzen $> f(max)$ von einem analogen Lowpass-(Highcut-) Filter herausgefiltert.
In Bezug auf diese Arbeit wird davon ausgegangen, dass das menschliche Gehör in der Lage ist, Frequenzen zwischen 20Hz und 20kHz wahrzunehmen [14]. Dadurch ergibt sich eine Mindestabtastrate von 44.100Hz oder ein Samplepunkt alle 22.7µs (Audio CD Redbook Standard [28]) [25].

4.2 Samplingtiefe

Die Samplingtiefe (auch Bittiefe oder engl. Bit-Depth) beschreibt in der vertikalen Auflösung (Amplitude) das Raster zur Transformation eines analogen in ein diskretes, digitales Signal. Es wird dabei auch von dem Dynamikumfang der Wandlung gesprochen. Wird von einer ω-Bit-Quantisierung des Eingangssignals ausgegangen, lassen sich durch das binäre System des Digitalrechners 2^w verschiedene Amplitudenwerte darstellen.

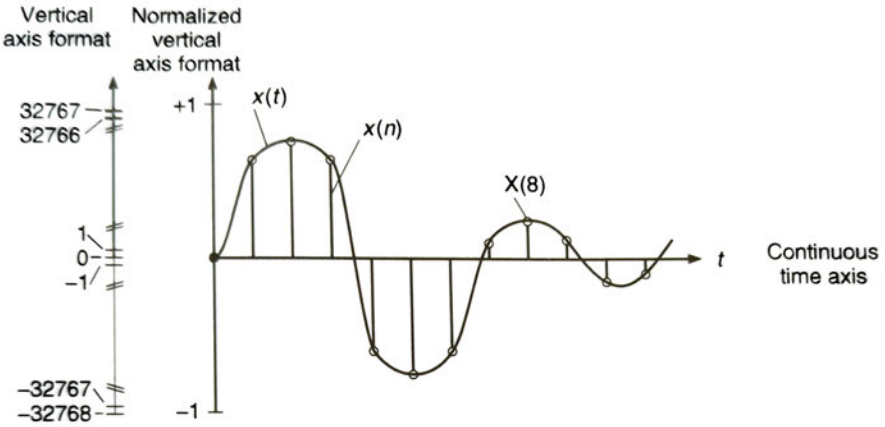

Abbildung 4.2: Darstellung der Bittiefe als 16-Bit-Wert und normalisierter Wertebereich [30].

Für die in Audioanwendungen gängigen Samplingtiefen ergeben sich dabei die folgenden Werte:

- 8 Bit - 256 Amplitudenwerte
- 16 Bit - 65.536 Amplitudenwerte
- 24 Bit - 16.777.216 Amplitudenwerte
- 32 Bit - 4.294.967.296 Amplitudenwerte

Auch wenn bereits Audioanwendungen existieren, die mit 32-Bit arbeiten, ist die 24-Bit Auflösung durch einen Dynamikumfang von 144db ausreichend. Daher wird in dieser Arbeit der heute übliche Standard von 24 Bit verwendet. Die anliegenden Werte werden nach der Wandlung von $-2^w - 1$ bis $2^w - 1$ -1 durch den Betrag von 2^w geteilt, um auf einen normalisierten Wertebereich zwischen -1 und 1 zu kommen (wie in Abbildung 4.2 dargestellt).

4.3 Fourier-Transformation

Im Jahr 1807 entdeckte der Mathematiker Jean Baptiste Joseph Fourier, dass sich eine periodische Funktion in ihre Linearkombinationen von Sinus- und Cosinusschwingungen zerlegen lässt. In der Audiodomäne ist diese Zerlegung von einem Zeit- zu einem Frequenzraum äußerst hilfreich, da diese es ermöglicht, komplexe Signale in einzelne Frequenzan-

teile (Grundton und Obertöne) zu zerlegen [20]. Diese Methode wird diskrete Fourier-Transformation (DFT) genannt. Analog dazu wird die Umwandlung eines Frequenzspektrums in eine zeitbasierte Darstellung als inverse diskrete Fourier-Transformation (IDFT) bezeichnet. Auch für die Faltung (engl. Convolution) von Audiosignalen mit Impulsantworten von verschiedenen Systemen, wie in diesem Fall HRTF-Aufnahmen, spielt die Fourier-Transformation eine entscheidende Rolle. Bei der in 4.1 gezeigten Formel ist es in der Praxis jedoch sehr aufwändig, die komplexen Fourier-Koeffizienten zu bestimmen. Die Zeitkomplexität der Laufzeit der diskreten Fourier-Transformation liegt bei $O(n^2)$.

$$x_p(t) = \frac{a_0}{2} + \sum_{k=1}^{\infty}[a_k \cos(2\pi k f_0 t) + b_k \sin(2\pi k f_0 t)] \tag{4.1}$$

Bei der Berechnung von Audiosignalen wird auf Grund ihrer Länge und des damit sehr hohen Zeitaufwands bei der Anwendung der Fourier-Transformation hauptsächlich die schnelle Fourier-Transformation (FFT) angewendet.

4.4 Fast-Fourier-Transformation

Die schnelle Fourier-Transformation bzw. die inverse schnelle Fourier-Transformation IFFT ist ein Verfahren, welches in der Lage ist, in einer linearithmischen Laufzeit von $O(nlog(n))$ eine Transformation zwischen der Frequenz- und Zeitdomäne durchzuführen. Die FFT wurde von Cooley und Tukey entwickelt und 1965 veröffentlicht [20]. Das FFT-Verfahren wird unter anderem bei den FIR-Filtern eingesetzt, die in der im Laufe dieser Arbeit entwickelten Anwendung für die Faltung mit den HRIR benötigt werden.
Der Unterschied in der Berechnung der FFT zur DFT liegt in der Speicherung und Wiederverwendung bereits berechneter Zwischenergebnisse und lässt sich wie folgt veranschaulichen. Bei der diskreten Fourier-Transformation wird eine Matrix mit einem Vektor multipliziert. Dazu müssen zur Berechnung von einer Frequenzstelle N Multiplikationen und $N-1$ Additionen durchgeführt werden. Demnach müssen für die N Frequenzstellen $N * (N + (N-1))$ Operationen ausgeführt werden, welche ungefähr einer Laufzeit von N^2 entsprechen.

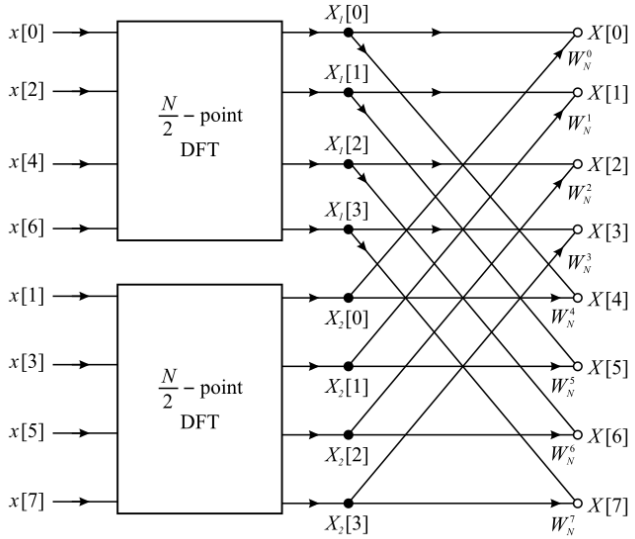

Abbildung 4.3: Aufspaltung einer 8-Punkte-DFT in zwei 4-Punkte-DFTs [20].

Wenn die in der folgenden Aufzählung genannten Voraussetzungen gegeben sind, kann durch ein Teile-und-Herrsche-Verfahren (engl. Divide and Conquer) die ursprüngliche $N * N$ Matrix in zwei Teilmatritzen der Größe $N/2$ zerlegt werden, wodurch sich die Rechenoperationen in der Größe $1/4$ verringern lassen.

- M ist eine $N * N$ Matrix
- Größe der Matrix entspricht einer Zweierpotenz

Wenn dieses Verfahren rekursiv bis zu einer Matrix der Größe 1 wiederholt wird, sinkt die Zeitkomplexität auf $O(nlog(n))$. Die erste Zerlegung einer 8-Punkte-DFT in zwei 4-Punkte-DFTs wird in Abbildung 4.3 gezeigt. Mit dem Verfahren der Fast-Fourier-Transformation können bei einer 1024-Punkte-DFT über 99 % der Rechenoperationen eingespart werden [20].

4.5 Filter

Die Funktionsweise eines digitalen Filters kann als eine mathematische Operation definiert werden, um selektiv Frequenzen eines Frequenzspektrums in der Amplitude anzuheben, abzusenken oder in der Phase zu bearbeiten [20]. Dazu besitzen Filter einen Sperrbereich, welcher das Signal absenkt und einen Durchlassbereich, welcher das Signal

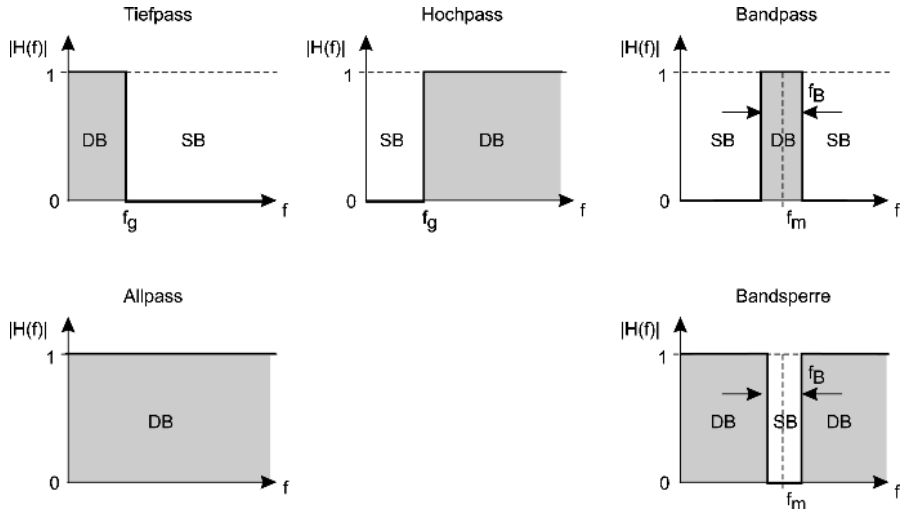

Abbildung 4.4: Filtertypen mit Durchlassbereich (Grau) und Sperrbereich (Weiß) [27].

unverändert passieren lässt. Filter besitzen in der digitalen Signalverarbeitung einen hohen Stellenwert. So finden sie Anwendung in der Breitbandübertragungstechnik, beim Rundfunk oder auch in der Mobilfunktechnik. In der Audiotechnik unterscheidet man zwischen fünf Basisfiltertypen, die in der folgenden Liste definiert sind und deren Klassifikation in Abbildung 4.4 gezeigt wird.

- Lowpass Filter (LP) - Lassen tiefe Frequenzen bis zur Grenzfrequenz f_g durch und bedämpfen Frequenzen oberhalb von f_g.
- Highpass Filter (HP) - Lassen hohe Frequenzen ab einer Grenzfrequenz f_g durch und bedämpfen Frequenzen unterhalb von f_g.
- Bandpass Filter (BP) - Lassen Frequenzen zwischen einer unteren Grenzfrequenz f_{gl} und einer oberen Grenzfrequenz f_{gh} durch.
- Bandreject Filter (BR) - Bedämpfen Frequenzen zwischen einer unteren Grenzfrequenz f_{gl} und einer oberen Grenzfrequenz f_{gh}.
- Allpass Filter - Lassen alle Frequenzen durch, modifizieren aber die Phase des Inputsignals.

Bei der Implementierung von Filtern in Computersystemen wird zwischen den zwei grundlegenden Filtertypen IIR und FIR unterschieden [30].

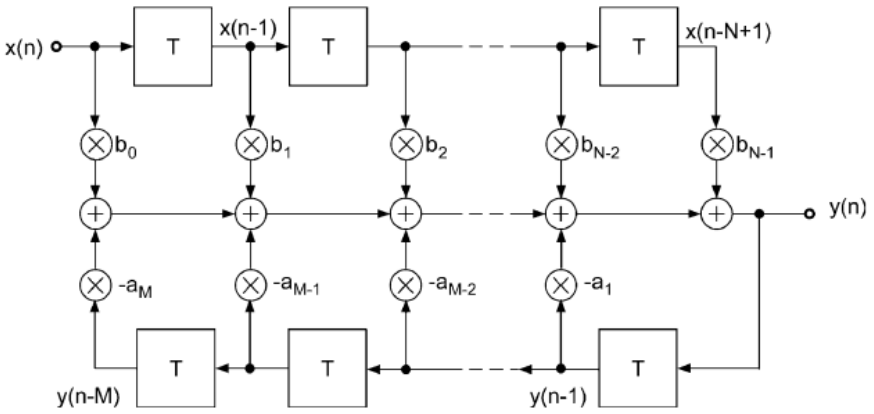

Abbildung 4.5: Blockschaltbild eines IIR-Filters der Ordnung $M > N - 1$ [27].

4.5.1 Infinite Impulse Response Filter

Infinite Impulse Response Filter (IIR) repräsentieren Filter, deren Impulsantwort eine unendliche Länge besitzen kann. Dies ist darauf zurückzuführen, dass nicht nur die Eingangswerte in die Berechnung einfließen, sondern auch, wie in Abbildung 4.5 zu sehen ist, die Ausgangswerte rekursiv für die Berechnung herangezogen werden.
IIR-Filter sind wie oben beschrieben rekursiv und garantieren dadurch keine Systemstabilität, d.h. sie können eine selbständige Eigenresonanz annehmen, die zu einer unendlichen Impulsantwort führen würde. Durch den deutlich geringeren Rechenaufwand gegenüber FIR-Filtern werden sie in der digitalen Audiobearbeitung generell dort eingesetzt, wo es nicht auf einen linearen Phasengang ankommt. Die Differenzengleichung in 4.2 veranschaulicht hierbei, dass die Faltungssumme nicht nur durch Abtastwerte des Eingangssignals, sondern auch durch rekursive Summierung des Ausgangssignals berechnet wird [30].

$$y(n) = \sum_{k=0}^{\infty} h(k)x(n-k) = \sum_{k=0}^{N-1} b(k)x(n-k) - \sum_{k=1}^{M} a(k)y(n-k) \tag{4.2}$$

4.5.2 Finite Impulse Response Filter

Finite Impulse Response Filter (FIR) besitzen mehr Filterkoeffizienten und sind in der Berechnung deutlich aufwändiger als IIR-Filter. Sie zeichnen sich aber durch eine fehlende

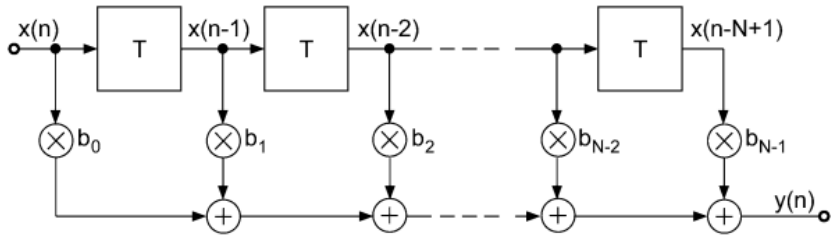

Abbildung 4.6: Blockschaltbild eines FIR-Filters der Länge N [27].

Rekursion und damit eine endliche Impulsantwort aus, die mit einer erhöhten Systemstabilität und einem linearen Phasengang und konstanter Gruppenlaufzeit einhergeht. Wie im Blockschaltbild 4.6 und durch die Differenzgleichung in 4.3 zu sehen ist, werden nur vorwärts gerichtete Operationen verwendet, die das Ausgangssignal einzig aus Werten des Eingangssignals berechnen.

$$y(n) = \sum_{k=0}^{N-1} b(k)x(n-k) \tag{4.3}$$

FIR-Filter arbeiten, indem sie die Input-Samples $x(n)$ mit den Samples einer Impulsantwort $b(n)$ multiplizieren und mit allen vorhergehenden Produkten zu einem Ausgangssignal $y(n)$ summieren. Diese Operation wird auch Faltung (engl. Convolution) genannt. Durch die Multiplikation des Inputsignals mit der Impulsantwort werden gleiche Frequenzen angehoben, während Frequenzen, die in der Impulsantwort nicht oder schwach vorhanden sind, bedämpft werden [30].

Ein Nachteil der FIR-Filter ist, dass durch die feste Länge der Impulsantwort die Samples des Inputsignals inkrementell für jeden Bearbeitungsschritt um eine Position verschoben werden müssen. Wenn der Audiobuffer als Array realisiert wurde, bedeutet dies ein Verschieben aller Werte des gesamten Arrays für jedes neue Audiosample. Die Menge der Datenoperationen auf dem Array ist ineffizient und würde den Bearbeitungsaufwand für eine Echtzeitanwendung übersteigen. Eine Abhilfe schafft hier der Ringbuffer, welcher als ein kreisförmiger Speicher angesehen werden kann, in dem lediglich ein Speicherwert überschrieben und ein Pointer auf die nächste Position gesetzt werden muss. Eine theoretische Analyse effizienter Datenstrukturen für FIR-Filter sind in der Literatur durch Dr. Steve Arar gegeben [8].

Nach Betrachtung der Grundlagen der digitalen Signalverarbeitung und dem Skizzieren verschiedener Filterdesigns sind die Voraussetzungen für die Entwicklung eines Algorithmus vorhanden, der die im vorhergehenden Kapitel eingeführten HRTF-Dateien für die Faltung mit einem Eingangssignal in einer Applikation nutzt.

5 Algorithmus

In diesem Kapitel wird anhand der gewonnenen Erkenntnisse über die räumliche Ortung von Schallereignissen und die digitale Signalverarbeitung ein Algorithmus entwickelt, der es ermöglicht, ein beliebiges Mono-Eingangssignal durch Faltung mit einer HRTF-Datei in einem virtuellen Raum zu positionieren und als binaurales Stereosignal auszugeben.

5.1 Entwurf

Im ersten Schritt werden dabei die Anforderungen an den Algorithmus definiert und in einer Blackboxansicht veranschaulicht. Um einen modularen Aufbau für die Implementierung zu gewährleisten, wird der Algorithmus dabei in zwei Iterationen beschrieben.

Basis-Algorithmus

Die funktionalen Anforderungen der Basisvariante sollen sicherstellen, dass ein Mono-Eingangssignal durch die Bearbeitung des Algorithmus in ein binaurales Stereo-Ausgangssignal überführt wird. In Abbildung 5.2 ist dabei die Blackboxansicht des Algorithmus aus Benutzersicht dargestellt. Die funktionalen Anforderungen sind dabei wie folgt definiert:

- Akzeptieren eines Mono-Eingangssignals als Datei oder als Mono-Audio-Input-Stream
- Faltung des Audiosignals mit einer im Vorfeld definierten HRIR WAVE-Datei
- Ausgabe einer binauralen Stereo-Audiodatei.

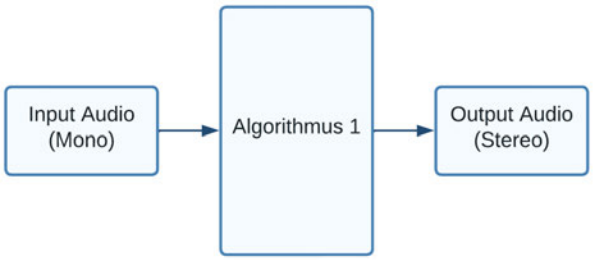

Abbildung 5.1: Benutzersicht der Basisvariante des Algorithmus mit Input/Output von Audiosignalen (Eigene Darstellung)

Erweiterter Algorithmus

Dieser Algorithmus setzt auf der Basisvariante auf und ist als Weiterentwicklung zu verstehen, die den Funktionsumfang des vorherigen Algorithmus erweitert. Dabei soll neben einer Möglichkeit, die HRIR-Daten aus WAVE-Dateien einzulesen, auch die Option bestehen HRIR-Daten, aus Dateien im Spatially Oriented Format for Acoustics (im Folgenden SOFA) zu verarbeiten (siehe Kapitel 5.2.2). Es soll dem Benutzenden ermöglicht werden, in einer grafischen Benutzeroberfläche (im Folgenden GUI) während der Laufzeit die Position und damit die zu verwendende HRIR zu ändern. Diese Positionsänderung bedingt in der Folge einen Audiostreaming-Ansatz, der es ermöglicht, die Audiodaten in Echtzeit zu bearbeiten und auszugeben. Die Blackboxansicht für den Algorithmus ist in Abbildung 5.2 beschrieben. Die funktionalen Anforderungen sind dabei wie folgt definiert:

- Akzeptieren eines Mono-Eingangssignals als Mono-Audio-Input-Stream
- Verarbeitung von HRIR-Daten als WAVE-Datei oder SOFA-Datei
- Wahl der Positionsdaten für die zu verwendende HRIR mittels eines GUI
- Faltung des Audiosignals mit der gewählten HRIR
- Ausgabe eines binauralen Stereo-Ausgangssignals.

Die nicht-funktionalen Anforderungen an den Algorithmus sind wie folgt definiert:

- Der Algorithmus soll auf verschiedenen Systemen und Betriebssystemen lauffähig sein
- Die Bearbeitungszeit muss eine Echtzeitbearbeitung des Audiosignals ermöglichen.

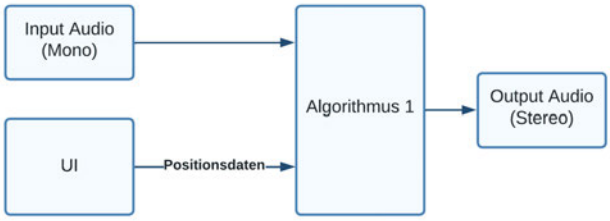

Abbildung 5.2: Useransicht der erweiterten Variante des Algorithmus mit Input/Output von Audiosignalen und Wahl der Position über eine GUI (Eigene Darstellung)

Die weitere Ausarbeitung des Algorithmus bezieht sich auf die erweiterte Variante, da für die Implementierung der Basisvariante des Algorithmus nur Teilbereiche umgesetzt werden müssen und nicht benötigte Anforderungen weggelassen werden können. Der Ablauf der einzelnen Teilschritte des Algorithmus wird in der folgenden Auflistung dargestellt.

1. Einlesen des Audiosignals als Monodatei oder Akzeptieren eines Audiostreams
2. Einlesen der HRIR Daten (WAVE oder SOFA)
3. Auswahl der korrekten HRIR anhand von Positionsdaten
4. Konvertierung der HRIR und des Audiosignals mittels FFT in die Frequenzdomäne
5. Multiplikation der HRTF mit dem Audiosignal (Faltung)
6. Konvertierung des gefalteten Audiosignals mittels IFFT in die Zeitdomäne
7. Ausgabe des Audiosignals als binaurales Stereosignal.

Die High-Level-Architektur in Abbildung 5.3 veranschaulicht die einzelnen Komponenten zur Berechnung der Stereosumme aus einem Mono-Eingangssignal und der HRIR-Daten. Dabei finden die Schritte 4 bis 6 der o.g. Auflistung in der Faltung statt. Der Algorithmus erhält von der Benutzer:in über das GUI im ersten Schritt eine Positionsangabe. Anhand dieser Position kann die korrekte HRIR-Datei aus einer Sammlung von WAVE-Dateien berechnet oder innerhalb einer SOFA-Datei die korrekte Speicheradresse bestimmt werden (siehe Kapitel 5.2.2). Nachfolgend wird der Audioinput mit der HRIR gefaltet und in einen Stereo-Ausgabebuffer geschrieben.

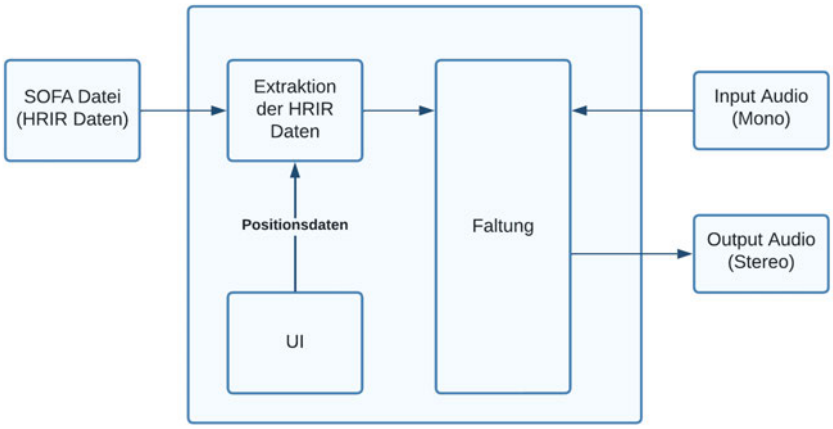

Abbildung 5.3: High-Level-Architektur des Algorithmus zur binauralen Positionierung von Audiosignalen (Eigene Darstellung)

5.2 Dateiformate für HRTF-Daten

Die in Kapitel 3.3 beschriebenen HRTF-Messungen werden in der Regel entweder als eigenständige Impulse Response in einer WAVE-Datei oder im SOFA-Dateiformat abgelegt. Da die Anzahl bei vollsphärischen Messungen bei einem Azimutwinkelabstand von 5 ° und einem Elevationwinkelabstand von 5 ° auf über 2.500 Messungen ansteigt und damit eine große Menge an WAVE-Dateien zur Folge hat (siehe CIPIC in Tabelle 3.1), wurde das SOFA-Dateiformat entwickelt, was die anfallenden HRIR bündelt und strukturiert mit allen relevanten Metainformationen in einer Datei speichern kann. In den nachfolgenden Unterkapiteln sollen die Funktionsweise und der Aufbau der einzelnen Dateiformate näher erläutert werden.

5.2.1 RIFF WAVE

Das WAVE-Dateiformat ist ein 1991 von Microsoft und IBM eingeführtes Containerformat für die Speicherung von Audiodaten im Little-Endian-Format und wurde ursprünglich für das Windows-Betriebssystem konzipiert. Durch die hohe Verbreitung ist mittlerweile jedoch fast jedes Computersystem in der Lage, mit WAVE-Dateien umzugehen. WAVE baut dabei auf dem Resource Interchange File Format (RIFF) auf, welches aus Datenblöcken, sogenannten Chunks, besteht. Diese einzelnen Chunks enthalten neben den reinen Pulse Code Modulation Daten (PCM) auch Metainformationen und können

ineinander verschachtelt sein. Wie in Abbildung 5.4 zu sehen ist, werden für eine WAVE-Datei, wie sie auch für die Speicherung von unkomprimierten HRIR-Daten genutzt wird, mindestens die Chunks RIFF, Format und Data benötigt. Der RIFF-Chunk gruppiert dabei den Format-Chunk, welcher die Metainformationen über Abtastrate, Kompression, Anzahl der Kanäle, Bit-Tiefe etc. enthält und den Data-Chunk, welcher neben einem Header die rohen Audiodaten in Form von signed-Integer-Werten beinhaltet [6].

5.2.2 Spatially Oriented Format for Acoustics

Das SOFA-Dateiformat wurde 2015 von der Audio Engineering Society als AES69-2015 standardisiert und speziell für die Speicherung von HRTF und HRIR entwickelt. Es basiert dabei auf dem Hierarchical Data Format (HDF) der fünften Version (HDF5), welches insbesondere von wissenschaftlichen Anwendungen für die strukturierte Speicherung großer Datenmengen verwendet wird [4].
SOFA soll das Problem adressieren, eine große Anzahl individueller HRIR einer Messreihe zu gruppieren und mit speziellen Metainformation wie Position, Länge der HRIR und Abtastrate etc. in einer Datei zu bündeln. Dazu wird die SOFA-Konvention wie in Abbildung 5.4 definiert. Der erste Block beinhaltet die HRIR-PCM-Daten und wird in Form eines eindimensionalen Arrays realisiert. Die Größe des Arrays wird dabei durch die Formel $M * K * N$ festgelegt, wobei M für die Anzahl der Messungen, K für die Anzahl der aufgenommenen Kanäle und N für die Anzahl der Samples pro Messung steht. Als Metainformation werden die Position der Quelle und des Ziels (Hörende:r) als eindimensionales Array abgespeichert. Die Positionsdaten werden dabei entweder als räumliche Polarkoordinaten (Radius r, Azimut δ, Elevation ϕ) oder als kartesische Koordinaten hinterlegt. Auch die Anzahl der Messungen sowie die Länge der HRIR werden in den Metainformation persistiert. Die Dimensionen C sind auf Grund des dreidimensionalen Koordinatensystems als Konstante mit dem Wert 3 zu belegen [7].

Extraktion der HRIR

Um die korrekte HRIR durch die vom Benutzer:in über das GUI festgelegte Position bestimmen zu können, wird eine Funktion benötigt, die SOFA-Dateien einlesen und die korrekte Position des Startpunkts im Speicher der angeforderten HRIR bestimmen und zurückgeben kann. Um dieses Ziel zu erreichen, wird eine Struktur wie in Abbildung 5.5 angelegt. Auf der rechten Seite befindet sich das IR-Array, welches die Audiodaten der

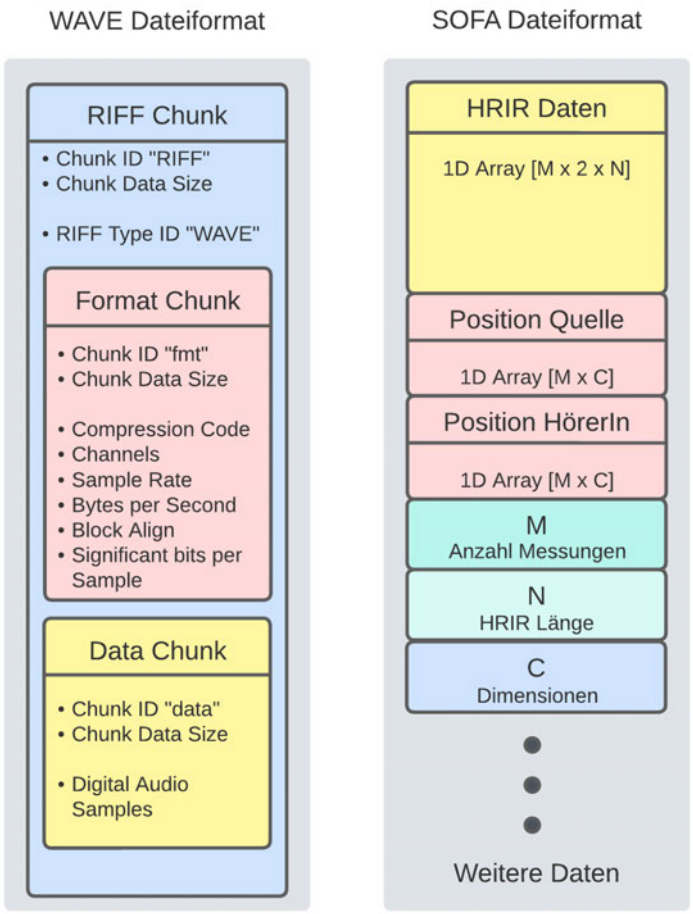

Abbildung 5.4: Links: Aufbau WAVE-Dateiformat mit den Chunks RIFF, Format, Data. Rechts: Aufbau SOFA-Dateiformat mit HRIR-Array und Metainformationen (Eigene Darstellung)

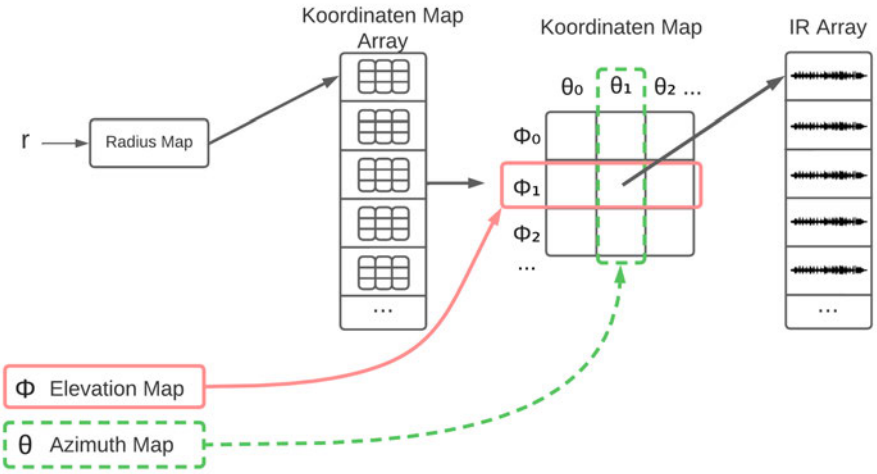

Abbildung 5.5: Extraktion der angewählten HRIR durch Dereferenzierung der Positionen zur Ermittlung der Speicheradresse (Eigene Darstellung nach [21])

HRIR enthält. Um den korrekten Index des Arrays bestimmen zu können wird in einer Radius Map der übergebene Radius gesucht und die korrekte Koordinaten-Map dereferenziert. Jeder Radius benötigt dabei eine eigene Koordinaten-Map, da es vorkommen kann, dass für unterschiedliche Radii eine unterschiedliche Anzahl an Elevation- und Azimut-Messungen und damit Einträgen in der Koordinaten-Map existieren. Im nächsten Schritt wird aus dem übergebenen Elevation-Parameter die Reihe der Koordinaten-Map bestimmt. Die Spalte wird analog dazu über den übergebenen Azimut-Wert bestimmt. Der so errechnete Index in der Koordinaten-Map gibt die Position der gesuchten HRIR im IR-Array an. So kann die Speicheradresse der gewünschten HRIR gefunden und zurückgegeben werden [21].

5.3 Faltung der Audiosignale

Der Prozess des Überlagerns der IR mit den Impulsen des Audiosignals und deren anschließende Addition im Zeitbereich wird Faltung genannt [25]. Die mathematische Integraloperation dafür ist wie folgt definiert:

$$y(t) = \int_{-\infty}^{\infty} x(\tau)h(t-\tau)d\tau \tag{5.1}$$

Dabei muss bei der Faltung zwischen der Zeitdomäne und der Frequenzdomäne unterschieden werden. In der Zeitdomäne wird das Signal Sample für Sample mit der HRIR multipliziert und das Ergebnis anschließend aufaddiert (siehe Kapitel 4.5.2). In der Frequenzdomäne hingegen werden die, durch eine Fourier-Transformation erhaltenen, Frequenzabbildungen der HRIR und des Audiosignals miteinander multipliziert. Diese Beziehung zwischen der Faltung in der einen und der Multiplikation in der anderen Domäne ist dual und besagt, dass auch eine Faltung im Frequenzbereich einer Multiplikation im Zeitbereich entspricht [20].

5.3.1 Anwendung der FIR-Filter

Die einfachste Anwendung der Faltung auf ein Audiosignal besteht darin, die in Kapitel 4.5.2 beschrieben FIR-Filter zu verwenden. Unter Einsetzung der HRIR als Koeffizientensatz in den FIR-Filter kann die Faltungsoperation in der Zeitdomäne wie in Abbildung 4.6 durchgeführt werden. Diese Art der Faltung reicht für die Basisvariante des vorgestellten Algorithmus aus, da durch die generierte Output-Datei keine Echtzeitbearbeitung gewährleistet werden muss. Da die Multiplikation Sample für Sample und die anschließende Summierung jedoch eine große Menge Rechenoperationen erfordern und dadurch ineffizient sind, kann bei IR mit Samplelängen über 128 Samples die Gefahr von Überlastung bei einer Echtzeitanwendung bestehen.

5.3.2 Anwendung der FFT und IFFT

Anders als die in Abbildung 4.6 dargestellte ineffiziente Multiplikation und anschließende Summierung des Signals in der Zeitdomäne, wird bei Audioanwendungen, die eine Echtzeitbearbeitung des Audiosignals sicherstellen müssen, die effizientere Multiplikation der

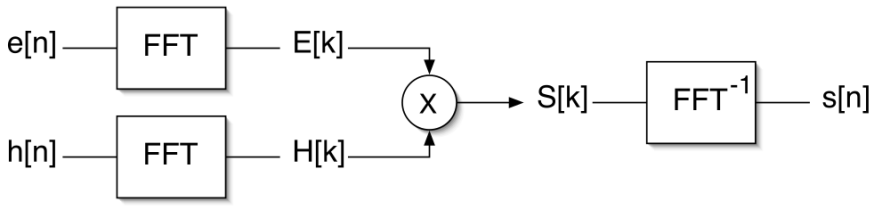

Abbildung 5.6: Faltung des Audiosignals im Frequenzspektrum nach vorheriger Konvertierung mittels der in Kapitel 4.4 beschriebenen FFT [24]

Frequenzabbildungen in der Frequenzdomäne favorisiert. Durch die Anwendung der in Kapitel 4.4 beschriebene FFT kann die HRIR und das Audiosignal aus der zeitlichen in die Frequenzdomäne überführt werden, wo die Frequenzabbildungen multipliziert werden. Nachfolgend wird das resultierende Frequenzbild mittels IFFT wieder in die Zeitdomäne überführt. Dieser Prozess wird in Abbildung 5.6 dargestellt.

5.3.3 Overlap and Add

Durch die Anwendung der im vorherigen Kapitel beschriebenen FFT zur Überführung der Faltungsoperation vom Zeit- in den Frequenzbereich wurde durch die gesteigerte Effizienz der Berechnung zwar eine Echtzeitbearbeitung des Audiosignals garantiert, jedoch muss für die FFT sowie die IFFT das gesamte Audiosignal vorliegen.
Dies kann bei langen Audiodateien zu einem Problem führen und ist bei einem Inputstream per Definition nicht möglich, da die Länge und die Werte der eintreffenden Samples vorher nicht bekannt sind.

Einen Lösungsansatz bietet das sog. Windowing des Eingangssignals. Dazu wird das Eingangssignal, wie in Abbildung 5.7 gezeigt, in mehrere gleichgroße Blöcke X der Länge $X(n)$ unterteilt. Auf diesen Blöcken wird dann, wie in Kapitel 5.3.2 beschrieben, separat die FFT, die Faltung und anschließend die IFFT zurück in die Zeitdomäne angewendet. Im folgenden Schritt werden die Blöcke in einen Overlap-and-Add-Buffer (im Folgenden OLA) geschrieben. Dabei gilt es zu beachten, dass durch die Faltung die Samplegröße des Augsgangsblocks ungleich der Samplegröße des Eingangsblocks ist und somit eine Überlappung (engl. Overlap) der einzelnen Ausgangsblöcke besteht.
Jeder Block wird beim Hinzufügen in den OLA-Buffer dabei höchstens um die Anzahl $X(n) * X$ Samples verzögert. Diese Verzögerung wird Hop H genannt und kann je nach gewähltem Windowing-Verfahren auch kleiner als die Länge des Audioblocks bestimmt

werden ($1 <= H <= X(n)$).
Je kleiner der Hop dabei gewählt wird, desto mehr Samples überlappen sich und werden bei weiteren Iterationen zur Berechnung herangezogen. Diese Überlappung erlaubt einen sanfteren Übergang der einzelnen Blöcke bei dynamischer Änderung der Audiodaten, wie z.B. dynamischen Wechsel der HRIR durch der Benutzer:in, erhöht dabei aber auch signifikant den Rechenaufwand. Aus diesem Grund sollte die Hop-Größe sorgfältig in Abhängigkeit mit dem Windowing-Verfahren gewählt werden und einen Kompromiss aus Geschwindigkeit und Qualität bieten.

Im letzten Schritt werden die einzelnen Ausgabeblöcke im OLA-Buffer addiert (engl. Add) und in einen Ausgabebuffer geschrieben. Diese Methode ermöglicht es dem Algorithmus auch bei langen oder unbekannten Audioeingangssignalen eine Echtzeitbearbeitung zu gewährleisten [25]. Die Größe des OLA-Buffers muss theoretisch als unendlich, jedoch mindestens so groß, definiert werden wie die zu verarbeitende Datei. Da dies in einer praktischen Anwendung, die mit einem Inputstream arbeitet, nicht realisierbar ist, wird er als Ringbuffer implementiert, (siehe Kapitel 4.5.2) der bei Erreichen der Buffergrenze den Pointer zurück auf Index 0 setzt [25].

5.3.4 Zero Padded Buffer

Wie im Kapitel 5.3.3 veranschaulicht, hat ein Block im OLA-Buffer nach der Fourier-Transformation und Faltung nicht die selbe Sampleanzahl und damit Länge wie der zugehörige Block im Inputbuffer. Durch die Implementierung des OLA-Buffers als Ringbuffer und das Rücksetzen des Pointers bei Erreichen der Buffergrenze besteht die Gefahr, dass die ersten Samples durch den Überhang korrumpiert werden. Dieses Verhalten wird auch Time Aliasing genannt und ist in Abbildung 5.8 dargestellt. Die Länge des gefalteten Signals wird dabei für die Signallänge $X(n)$ und die HRIR-Länge $H(n)$ durch die Formel $X(n) + H(n) - 1$ definiert. Eine Möglichkeit das Time Aliasing zu verhindern, besteht darin, die Buffer mindestens mit einer Größe von $X(n) + H(n) - 1$ zu initialisieren. Diese Methode ist in Abbildung 5.9 beschrieben. Meist wird jedoch die nächst höhere Zweierpotenz gewählt. Die Buffer werden dann mit Nullen initialisiert, um bei nicht vollständiger Belegung keine Störgeräusche zu verursachen [30].

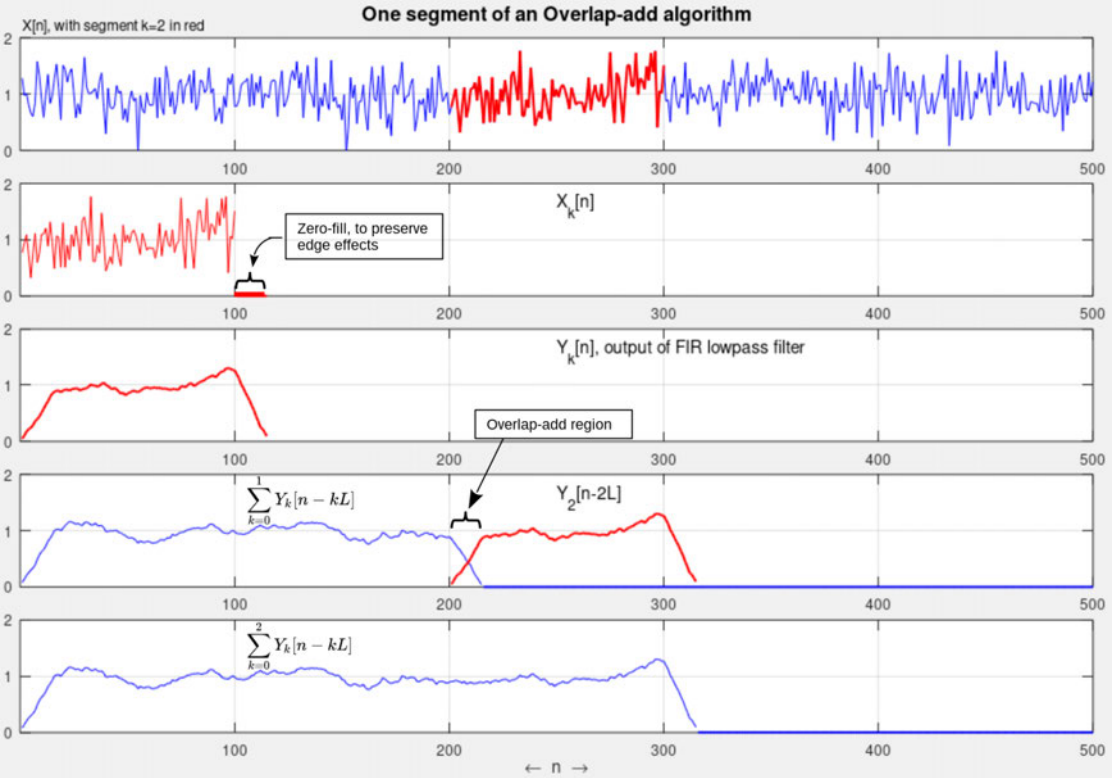

Abbildung 5.7: Fünf Schritte eines kontinuierlichen Signals, welches mittels Overlap and Add in Blöcke zerlegt, bearbeitet und anschließend wieder aufaddiert wird [3].

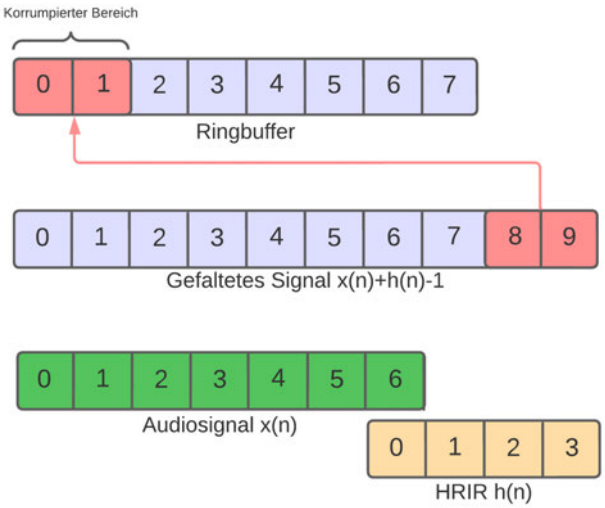

Abbildung 5.8: Durch eine zu geringe Buffergröße werden die ersten Samples überschrieben, was sich Time Aliasing nennt (Eigene Darstellung)

Diese Technik heißt Fast Convolution und wird durch die folgenden Schritte definiert:

- Zero Padding für das Signal und die HRIR auf die Länge $X(n) + H(n) - 1$
- Ausführen der FFT auf HRIR und Audiosignal
- Multiplikation der Frequenzspektren
- Ausführen der IFFT auf dem resultierenden Frequenzbild

In diesem Kapitel wurde ein Algorithmus entwickelt, der einen Inputstream und Positionsdaten akzeptiert und in der Lage ist, einen binauralen Outputstream bereitzustellen. Für dessen Beschreibung wurde ein Überblick über verschiedene Dateiformate gegeben, in dem die HRIR-Daten vorliegen können und aufgezeigt, wie die korrekte HRIR ermittelt und extrahiert werden kann. Des Weiteren wurde auf die Faltung in der Frequenzdomäne eingegangen und Ansätze zur Lösung der Probleme bei der Echtzeitbearbeitung mittels OLA- und Zero-Padded-Buffer behandelt. In Abbildung 5.10 ist der gesamte Ablauf des Algorithmus beschrieben.

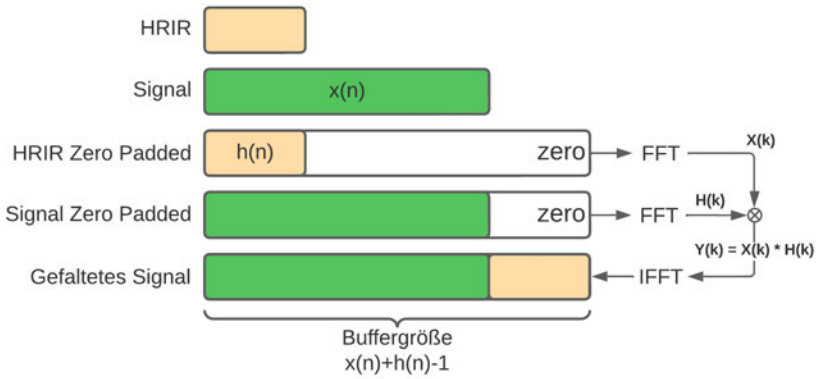

Abbildung 5.9: Fast-Convolution-Algorithmus zur Angleichung der Buffergrößen (Eigene Darstellung)

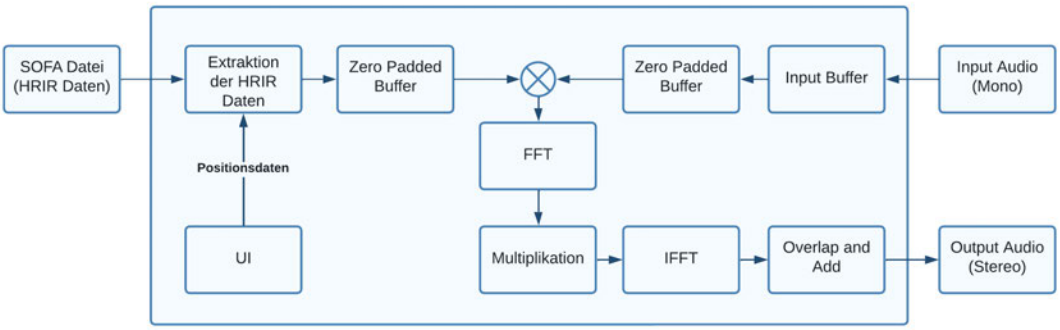

Abbildung 5.10: Diagrammatische Darstellung des Algorithmus zur Erstellung eines binauralen Stereo-Audiostreams aus einem Mono-Audio-Input-Stream (Eigene Darstellung)

6 Anwendung

Nachfolgend wird die Softwareanwendung zur Umsetzung des in Kapitel 5 beschriebenen Algorithmus definiert. Dabei wird auf die Besonderheiten von Audioprogrammierung sowie Methoden zur Einhaltung der Echtzeitverarbeitung von Audiosignalen eingegangen.

6.1 Anforderungen

Um die Anforderungen zu definieren, muss in erster Instanz festgelegt werden, auf welchen Zielsystemen die Anwendung zum Einsatz kommen soll. In Abhängigkeit dazu müssen die verschiedenen Schnittstellen definiert werden, mit der die Anwendung kommunizieren kann. Betriebssysteme wie Linux, macOS, Windows auf stationären Rechnern, aber auch iOS und Android auf Mobilgeräten bieten mit systemspezifischen APIs Schnittstellen zu den Treibern der Audiohardware an. Eine weitere Anwendungsumgebung für Audioprogramme sind Digital Audio Workstations (im Folgenden DAW). Dort werden die Applikationen als Plugin geladen und müssen mit den verschiedenen Schnittstellen der konkurrierenden Hersteller kompatibel sein. Die gängigsten Plugin-Formate sind hierbei AAX von Avid für Pro Tools, AU von Apple für Logic und VST von Steinberg für Cubase oder Nuendo.

Da für die zu entwickelnde Audioanwendung eine plattformübergreifende Kompatibilität angestrebt wird, liegt die Verwendung eines Frameworks nahe, das die Kommunikation mit den Schnittstellen vereinheitlicht. Dazu wird das im Kapitel 6.1.2 vorgestellte JUCE-Audioframework eingesetzt.

Nicht nur die Schnittstellen zu anderen Systemen müssen im Vorfeld betrachtet werden. Auch in sich ist eine Audioanwendung ein komplexes Zusammenspiel verschiedener Bausteine. Neben einem GUI benötigt die Audioanwendung eine Methode, die in Abhängigkeit der Abtastrate und der gewählten Buffergröße, die Audiodaten in einer fest

Buffer size	**Buffer length** @ sample rate 44.1 kHz	**Buffer length** @ sample rate 96 kHz
32 samples	0.73 ms	0.33 ms
64 samples	1.45 ms	0.66 ms
128 samples	2.90 ms	1.33 ms
…	…	…
1024 samples	23.2 ms	10.7 ms

Abbildung 6.1: Buffergrößen und die resultierende Latenz in Computersystemen [18]

definierten Zeit der Audiohardware zur Verfügung stellt. Dabei wird von der sogenannten harten Echtzeitverarbeitung (engl. Hard-Realtime-System) gesprochen. Anders als Non-Realtime-Systeme, die in der Regel versuchen eine möglichst hohe Effizienz und Durchsatz zu erreichen, und Soft-Realtime-Systeme, die zwar in einer bestimmten Zeit ein Ergebnis liefern müssen, dieses aber auch gelegentlich verfehlen dürfen, ist bei einem Hard-Realtime-System ein einmaliges Überschreiten der Berechnungszeit unzulässig und wird als Systemfehler gewertet [17]. Das folgende Beispiel soll aufzeigen, weshalb es unerlässlich ist, bei Audioanwendungen die Einschränkungen der harten Echtzeitberechnung zu erfüllen.

Die in Kapitel 4 beschriebene Abtastung des zeitkontinuierlichen Audiosignals und dessen Konvertierung in ein zeitdiskretes Signal mit einer Abtastrate von 44.100 Hz würde alle 22.7 µs einen Callback der Audiohardware bedeuten. Da diese hohe Frequenz an Methodenaufrufen technisch nicht realisierbar ist, werden die Audiodaten in einem Buffer zwischengespeichert und als Paket an die Audiohardware übergeben. Die Größe dieses Buffers variiert in der Regel zwischen 32 bis 1024 Samples und legt die somit in Abbildung 6.1 dargestellten Latenzen fest. Hierbei muss beachtet werden, dass ein interaktives System eine Latenz von unter 10 ms anstreben sollte, um keine fühlbare Verzögerung zu verursachen.
Wenn davon ausgegangen wird, dass bei einer Buffergröße von 128 Sampeln und einer Abtastrate von 44.100 Hz die Audiohardware alle 2.90 ms einen Callback-Methodenaufruf durchführt, um einen neuen Buffer an Audiosamples verarbeiten zu können, wird bei Überschreiten dieser Zeit der gesamte Buffer von 128 Samplen verworfen. Die Folgen wären Aussetzer im Audiosignal und inakzeptable Störgeräusche wie Klicks und Knacken.

6.1.1 Wahl der Programmiersprache

Die im vorhergehenden Kapitel vorgestellte Anforderung der harten Echtzeitberechnung von Audioinformationen schränkt die Wahl der Programmiersprache ein. So muss eine Sprache gewählt werden, die schnell und hardwarenah ist, aber wegen der Komplexität der Anwendung trotzdem Abstraktionen, wie das objektorientierte Programmieren ermöglicht. Diese Kriterien erfüllt die Programmiersprache C++, welche ab 1979 von Bjarne Stroustrup als Erweiterung der Programmiersprache C entwickelt wurde. Sie zeichnet sich durch eine hohe Abstraktionsebene, ebenso wie durch ein Low-Level-Memorymanagement und eine daraus resultierende hohe Geschwindigkeit aus und ist in der Audioprogrammierung sehr verbreitet [9].

6.1.2 JUCE-Audioframework

Wie im Kapitel 6.1 ausgeführt, muss eine Audioanwendung auf verschiedenen Systemen mit unterschiedlichen APIs kommunizieren können. Des Weiteren setzt die harte Echtzeitverarbeitung der Audiosignale einen hohen Kenntnisstand der zugrundeliegenden Datenstrukturen und Algorithmen voraus, um diese einhalten zu können.

Das JUCE-Framework (Jules' Utility Class Extensions) ist ein Open-Source C++ Framework, welches im Jahr 2004 von Julian Storer entwickelt wurde und sich zum Ziel gesetzt hat, die Audioprogrammierung in den o.g. Punkten zu vereinfachen und Anwendern einen Rahmen zur Entwicklung von professionellen Audioanwendungen zu schaffen. Dabei lässt sich der Code ohne Anpassungen für alle gängigen Betriebssysteme wie Linux, macOS und Windows, aber auch für mobile Plattformen, wie iOS und Android kompilieren. Darüber hinaus werden auch Wrapperklassen für die meisten Plugin-Typen der großen DAWs (VST, AU, AAX) bereitgestellt [16].

Neben Möglichkeiten zum Erstellen eines GUI, arbeiten mit Netzwerk, Multithreading und Verarbeiten verschiedener Dateitypen, bietet JUCE vor allem die Möglichkeit effektiv und effizient mit Audiosignalen umzugehen. Zu diesem Zweck existieren viele Helferklassen, von denen abgeleitet werden kann oder deren Methoden zum Berechnen und Manipulieren von Audiosignalen genutzt werden können.
Ein JUCE-Projekt besteht neben den Framework-Dateien aus den zwei modifizierbaren C++ Dateien `PluginEditor.cpp` und `PluginProcessor.cpp`. PluginEditor ist dabei für das GUI zuständig und läuft auf einem low-priority-thread. PluginProcessor verwaltet

das DSP. Einstiegspunkt ist dabei die Methode `ProcessBlock(juce::AudioBuffer<float>& buffer)`, die einen Audiobuffer mit 32-Bit-Float-Werten als Parameter erhält und die Werte nach der Abarbeitung in den Outputbuffer schreibt. Diese Methode läuft auf einem eigenen Audiothread mit hoher Systempriorität und unterliegt der o.g. harten Echtzeitverarbeitung.

6.2 Anwendungsdesign

Im Kapitel 5.1 wurden die einzelnen Schritte des Algorithmus definiert. Diese sollen nachfolgend in einer Applikation umgesetzt werden. Da eine vollumfängliche Implementierung des Algorithmus inklusive der Einarbeitung in die benötigte Programmiersprache und des JUCE-Frameworks den Rahmen dieser Arbeit übersteigen würde, wird hier auf eine Implementierung der Grundfunktionalität zur binauralen Positionierung von Schallereignissen eingegangen [12]. Zu diesem Zweck wird das Grundgerüst der Beispielapplikation `ConvolutionDemo` genutzt, welche dem JUCE-Framework beiliegt. Diese bietet neben einer Möglichkeit zum Laden von Audiodateien ein Drop-Down-Menu, um verschiedene IR für die Faltung mit dem Audiosignal zu wählen. Im Folgenden werden die einzelnen Schritte des Algorithmus und deren Implementierung aufgezeigt.

Wahl der Audiodatei

In der Datei `DSPDemos_Common.h` wird mittels der Methode `openFile()` ein `FileChooser` Objekt instanziiert, welches in der Lage ist, eine beliebige Audiodatei des Formates `wav, mp3, aif` einzulesen und zu verarbeiten.

Einlesen der HRIR-Daten

Die HRIR-Daten werden in der Datei `ConvolutionDemo.h` eingelesen. Dazu wurde eine Auswahl an IR der in Kapitel 3.4 beschriebenen HRTF-Datenbank KEMAR verwendet. Die verwendeten Optionen und die dazugehörigen Azimut- und Elevation-Winkel sind in Tabelle 6.1 aufgelistet und können über ein Drop-Down-Menu in der Applikation gewählt werden. Die Methode `updateParameters()` in der Datei `ConvolutionDemo.h` beinhaltet dabei die Dateinamen der einzulesenden HRIR-Dateien. Im ersten Schritt wird der Dateiname an die Methode `createAssetInputStream()` in der Datei `DemoUtilities.h` übergeben und

Option	Azimut	Elevation	Radius
Bypass	—	—	—
Front	0°	0°	1m
Back	180°	0°	1m
Left	270°	0°	1m
Right	90°	0°	1m
Up	0°	90°	1m
Down	0°	-40°	1m
Left-Up	320°	60°	1m
Right-Up	50°	60°	1m
Left-Back-Down	220°	-30°	1m
Right-Back-Down	140°	-30°	1m
Left-Back	220°	0°	1m
Right-Back	140°	0°	1m

Tabelle 6.1: Wählbare Faltungsoptionen und die verwendeten Elevation und Azimut-Winkel

ein `InputStream` erzeugt. Im weiteren Verlauf wird ein `AudioBuffer<float>` erzeugt und der `InputStream` sukzessive in den `AudioBuffer` kopiert.

Faltung des Audiosignals mittels der JUCE-Convolution-Klasse

Die nachfolgende Konvertierung in das Frequenzspektrum sowie die Faltung des Audiosignals mit der HRIR und Rückkonvertierung in die Zeitdomäne kann auf zwei unterschiedliche Weisen realisiert werden. Zum Einen bietet das JUCE-Framework eine `juce::dsp::Convolution` Klasse, die mit der Methode `loadImpulseResponse()` einen Pointer zu einer Speicheradresse oder einen AudioBuffer akzeptiert. Im Anschluss kann mit der Methode `process()` das Audiosignal mit der zuvor geladenen HRIR gefaltet werden. Dabei transformiert die Klasse das Audiosignal und die HRIR intern durch eine FFT erst in den Frequenzbereich, um eine effektive Bearbeitung zu ermöglichen. Des Weiteren werden bei der FFT und IFFT die in Kapitel 5.3.3 und 5.3.4 beschriebenen Probleme durch den Einsatz eines Zero Padded Buffer und eines Overlap-and-Add-Buffer adressiert und gelöst.

Konvertierung der HRIR und des Audiosignals mittels FFT in die Frequenzdomäne

Einen anderen Ansatz stellt die explizite Überführung des Audiosignals und der HRIR unter Zuhilfenahme der Klasse `FFT` und der Methode `perform()` in das Frequenzspektrum dar. Dabei muss jedoch beachtet werden, dass die in Kapitel 5.3.3 und 5.3.4 beschriebenen Probleme durch eigene Implementierungen eines Zero Padded Buffer und eines Overlap-and-Add-Buffer gelöst werden müssen.

Multiplikation der HRTF mit dem Audiosignal (Faltung)

Das, durch die FFT in die Frequenzdomäne transformierte Audiosignal und die ebenfalls transformierte HRTF können nun multipliziert und in einen temporären Buffer geschrieben werden.

Konvertierung des gefalteten Audiosignals mittels IFFT in die Zeitdomäne

Für die Rückkonvertierung in die Zeitdomäne wird erneut die Klasse `FFT` mit der Methode `perform()` genutzt. Durch den optionalen Inverse-Parameter kann dabei statt einer FFT eine IFFT ausgeführt werden.

Ausgabe des Audiosignals als binaurales Stereosignal

Im letzten Schritt wird über den Buffer, in dem das gefaltete Audiosignal gespeichert ist iteriert und das Ergebnis in den Outputbuffer geschrieben.

In diesem Kapitel wurden die Probleme und Lösungsansätze bei Audioprogrammierung und speziell bei der Implementierung des in Kapitel 5.1 vorgestellten Algorithmus erörtert. Dabei bildet die entwickelte Anwendung die Grundfunktionalität des Algorithmus ab und ist in der Lage, beliebige Audiosignale mit 12 verschiedenen HRIR zu falten und somit 12 unterschiedliche Hörereignisorte zu produzieren.

7 Evaluation

In diesem Kapitel wird der in Kapitel 6 implementierte Algorithmus evaluiert. Zu diesem Zweck werden 24 binaurale Audiodateien generiert und anonymisiert einer Gruppe verschiedener Tester:innen zur Verfügung gestellt. Durch eine selektive Richtungsbestimmung der verschiedenen Schallereignisse sollen diese die Ortbarkeit des Algorithmus bewerten. Die somit gewonnen Daten werden im nachfolgenden Schritt ausgewertet und interpretiert.

7.1 Aufbau

Um eine aussagekräftige Datenbasis zu erhalten, müssen im Vorfeld diverse binaurale Audiodateien erzeugt werden, die Schallquellen an unterschiedlichen Positionen beinhalten. Dabei muss sowohl die interaurale Lokalisation auf der Frontal- und der Horizontalebene untersucht werden, als auch die monoaurale Lokalisation auf der Medianebene. Auch die Unterschiede bei der Ortung von Sprache im Gegensatz zu Musik bzw. breitbandigen Signalen sollen innerhalb dieses Experiments betrachtet werden. Zu diesem Zweck wurden die in Tabelle 6.1 dargestellten 12 Positionen sowohl mit Sprache, als auch mit einer Akusikgitarre prozessiert und der binaurale Output als WAV- und MP3-Datei exportiert.

Im Anschluss wurde mittels Vue.js eine Webapplikation entwickelt [13], die es ermöglicht, die Testdaten anonymisiert an die Tester:innen auszuspielen. Innerhalb der Applikation werden die ausgewählten Positionen zusammen mit der Zuordnung der Audiodatei in einem Array abgelegt und am Ende des Testprozesses mittels einer REST-Anfrage an das Backend in einer CSV-Datei abgelegt, die anschließend ausgewertet und visualisiert wird.

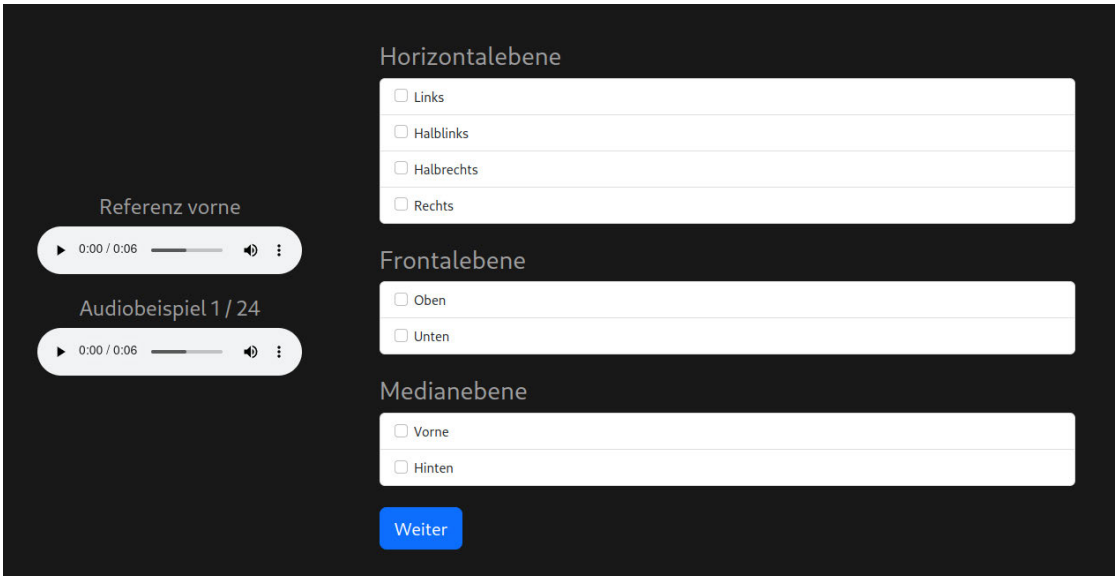

Abbildung 7.1: Oberfläche der Webapplikation zur Evaluierung des Algorithmus (Eigene Darstellung)

7.2 Durchführung

Die Webapplikation wird einer vorher festgelegten Gruppe von 12 Tester:innen zugänglich gemacht. Um eine fehlerfreie Durchführung zu gewährleisten wird vor Beginn des Tests durch einen Hinweistext auf die Einhaltung von bestimmten Richtlinien hingewiesen. Diese Vorgaben sollen sicherstellen, dass alle notwendigen Voraussetzungen erfüllt sind, um eine korrekte Durchführung der Evaluation zu gewährleisten. Dazu wird überprüft, dass die testende Person die Evaluierung auf Kopfhörern durchführt und durch ein Testsignal auf dem linken Ohr wird zusätzlich eine Vertauschung der Kopfhörerkanäle und eine damit fehlerhafte Lokalisierung auf der Horizontalebene ausgeschlossen.

Die testende Person bekommt aufeinanderfolgend 24 Audiodateien vorgespielt. Dabei enthalten die ersten 12 Audiodateien ein Sprachsignal und die anderen 12 Audiodateien ein Musiksignal. Wie in Abbildung 7.1 zu sehen ist, wird neben der Möglichkeit, die zu bewertenden Audiodatei erneut abzuspielen, eine Referenzaufnahme zur Verfügung gestellt, die ein Schallereignis mit $0°\varphi$ und $0°\delta$ beinhaltet. Die testende Person hat die Möglichkeit auf der Horizontalebene zwischen den Positionen links, halblinks, halbrechts und rechts zu wählen. In der Frontalebene stehen die Positionen oben und unten zur Verfügung. In der Medianebene werden die Auswahlmöglichkeiten vorne und hinten gelistet.

7.3 Ergebnis und Interpretation

Nach erfolgreichem Abschluss der Testreihe bilden 12 vollständige Datensätze die Grundlage für die in diesem Kapitel vorgenommenen Auswertungen. Durch eine Befragung der testenden Personen wurde ermittelt, dass die Unterscheidung in der Horizontalebene zwischen den Positionen links und halblinks, sowie rechts und halbrechts zu Verwirrung führte. Als Schlussfolgerung werden die Positionspaare zu jeweils einer Position links bzw. rechts zusammengefasst. Bei der Auswertung wird der Fokus auf die für jede Aufnahme relevanten Lokalisationsebenen gelegt und zusätzlich ausgewählte Parameter auf anderen Ebenen vernachlässigt. Im Folgenden werden die Auswertungen auf Basis der mit Sprache gefalteten Audiodateien vorgenommen. Im Anschluss werden dann die Unterschiede zwischen der mit Sprache und der mit dem Musiksignal prozessierten Audiodateien evaluiert.

Zusammenfassung der Ergebnisse

Im ersten Durchlauf wurde den Tester:innen ein Schallereignis auf der linken Seite mit 270°φ und ein Schallereignis auf der rechten Seite mit 90°φ vorgespielt. Wie in Abbildung 7.2 zu sehen, wurde das auf der linken Seite plazierte Schallereignis von allen Tester:innen ausnahmslos links verortet. Analog dazu wurde das auf der rechten Seite plazierte Schallereignis von allen Tester:innen rechts verortet. Damit ist die Links-Rechts-Ortung auf der Horizontalebene fehlerfrei erfolgt.

Im zweiten Durchlauf wurde den Tester:innen ein Schallereignis links oben mit 310°φ und 60°δ und ein Schallereignis rechts oben mit 50°φ und 60°δ vorgespielt. In der Auswertung in Abbildung 7.3 wird dabei die Ortung auf der Frontalebene gezeigt. Dabei ist auch hier die Links-Rechts-Ortung nahezu fehlerfrei erfolgt. Einzig eine Zuordnung rechts ist in der Auswertung von einer testenden Person nicht vorgenommen worden. Bei der Oben-Unten-Ortung haben auf der linken Seite etwas weniger als die Hälfte der Testpersonen eine korrekte Zuordnung vorgenommen. Lediglich eine Person hat die Schallquelle fälschlicherweise von unten geortet. Auf der rechten Seite sind ein Drittel richtige und keine falsche Zuordnung erfolgt.

Im dritten Durchlauf wird die Bestimmung auf der Medianebene untersucht. Dazu wurde den Tester:innen ein Schallereignis von vorne mit 0°φ und ein Schallereignis von hinten

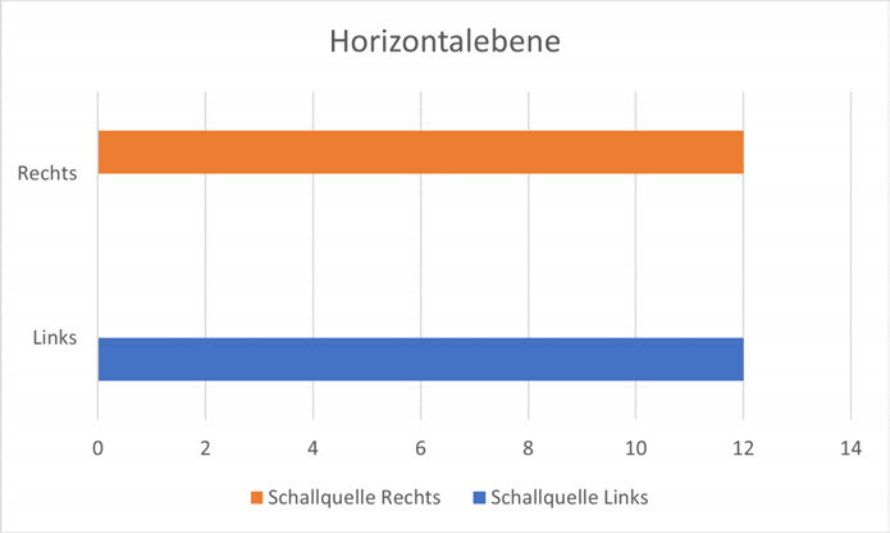

Abbildung 7.2: Wahrgenommener versus tatsächlicher Hörereignisort auf der Horizontalebene (Eigene Darstellung)

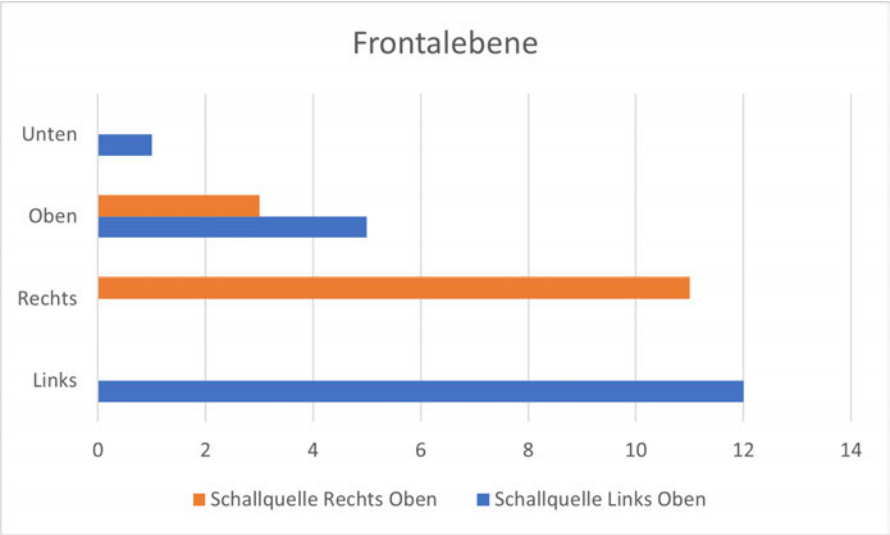

Abbildung 7.3: Wahrgenommener versus tatsächlicher Hörereignisort auf der Frontalebene (Eigene Darstellung)

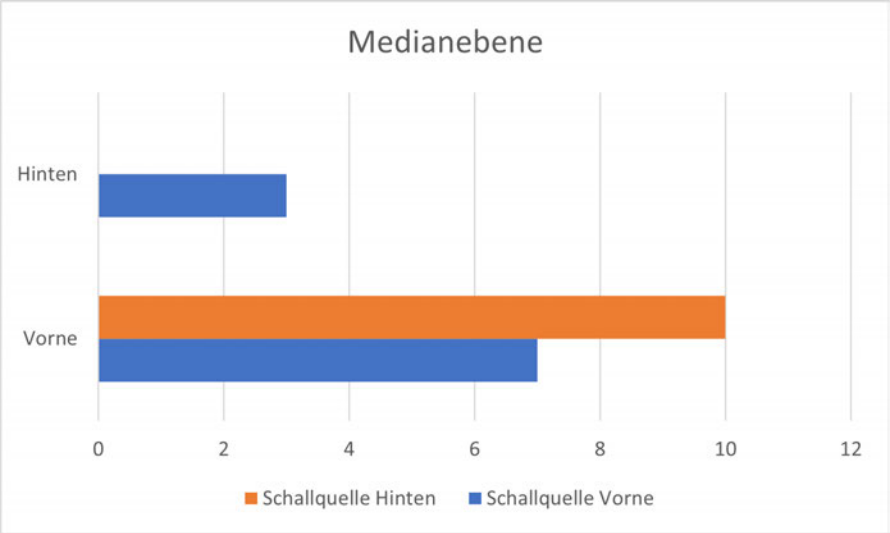

Abbildung 7.4: Wahrgenommener versus tatsächlicher Hörereignisort auf der Medianebene vorne / hinten (Eigene Darstellung)

mit 180°φ vorgespielt. In Abbildung 7.4 ist zu erkennen, dass sieben Personen die Schallquelle vorne richtig zuordnen konnten. Drei Personen haben das Schallereignis von hinten wahrgenommen. Die Schallquelle, die im Raum hinter der hörenden Person angeordnet wurde, wurde hingegen von zehn Personen fälschlicherweise vorne lokalisiert. Zwei Personen haben keine Zuordnung auf der Medianebene vorgenommen.

In Abbildung 7.5 wird ebenfalls die Bestimmung in der Medianebene untersucht. Bei dieser Auswertung wurden die Schallereignisse jedoch oberhalb und unterhalb der hörenden Person positioniert. Die Hälfte der Testpersonen hat die untere Schallquelle richtig verortet, wobei es nur ein Viertel bei der oberen Schallquelle korrekt bestimmen konnten. Auch wurde die obere Schallquelle fälschlicherweise drei Mal hinter der hörenden Person verortet. Bei der unteren Schallquelle ist die falsche Zuordnung hinter der hörenden Person vier Mal erfolgt.

In Abbildung 7.6 wurde das Schallereignis rechts, unten, hinter bzw. links, unten, hinter der hörenden Person positioniert. Auch in diesem Fall lagen elf der 12 Testpersonen mit ihrer Links-Rechts-Ortung richtig, wobei vier Personen auf der rechten und drei Personen auf der linken Seite die Oben-Unten-Ortung korrekt bestimmen konnten. Bei der Vorne-Hinten-Ortung waren es links sechs und auf der rechten Seite sieben von 12 Personen, die eine korrekte Zuordnung bestimmen konnten. Eine Person hat fälschlicherweise die Zuordnung auf der linken Seite von vorne bestimmt.

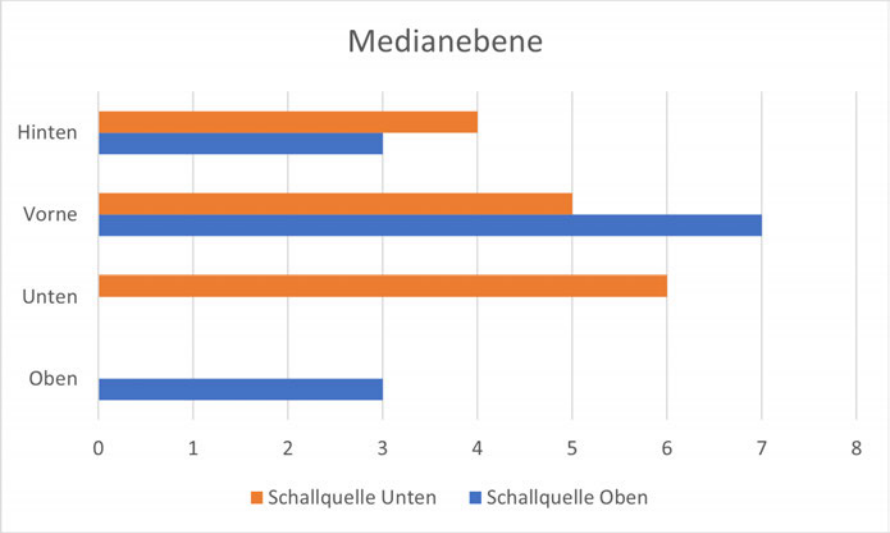

Abbildung 7.5: Wahrgenommener versus tatsächlicher Hörereignisort auf der Medianebene Oben / Unten (Eigene Darstellung)

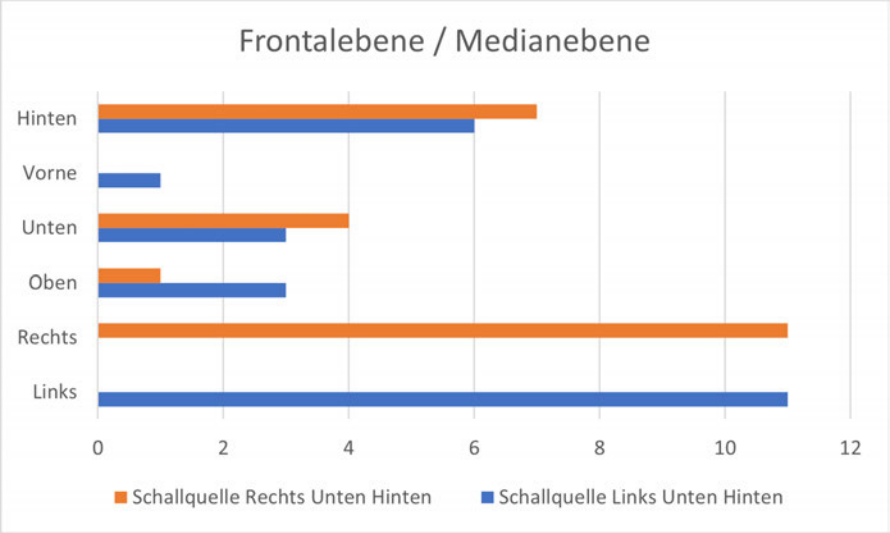

Abbildung 7.6: Wahrgenommener versus tatsächlicher Hörereignisort auf der Frontalebene und der Medianebene (Eigene Darstellung)

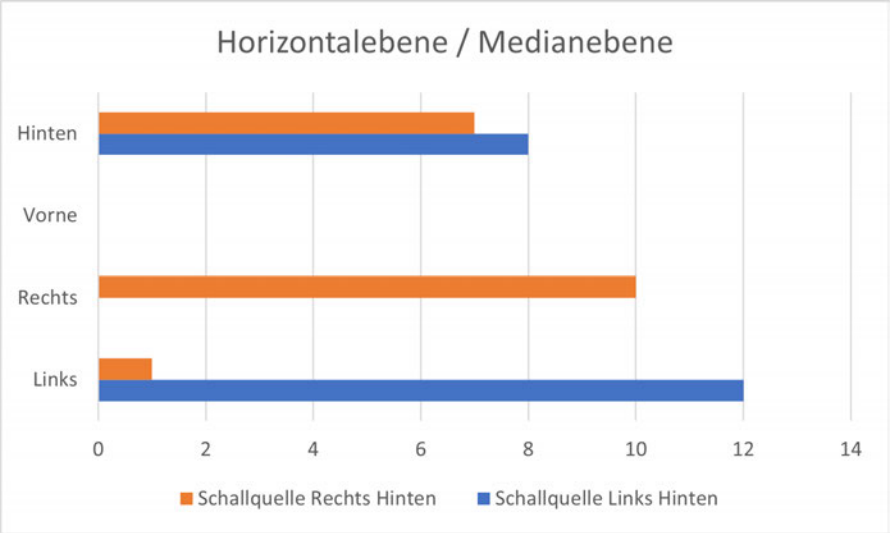

Abbildung 7.7: Wahrgenommener versus tatsächlicher Hörereignisort auf der Horizontalebene und der Medianebene (Eigene Darstellung)

In Abbildung 7.7 wird die Lokalisation eines Schallereignisses auf der Horizontalebene und der Medianebene untersucht. Dabei wird das Schallereignis rechts, hinten bzw. links, hinten angeordnet. Die linke Lokalisation wurde von allen 12 Testpersonen richtig erkannt. Auf der rechten Seite nahm eine Person das Schallereignis fälschlicherweise auf der gegenüberliegenden Seite wahr. Die Lokalisation auf der Medianebene konnten links acht und rechts sieben Personen richtig zuordnen. Dabei gab es keine gegensätzlichen Zuordnungen.

In Abbildung 7.8 wird die Ortung von Sprache und Musiksignalen in der Frontalebene verglichen. Dabei lässt sich feststellen, dass es keine starken Abweichungen zwischen den Signalarten gibt. Die Links-Rechts-Ortung ist bei beiden Signalen fast fehlerfrei und auch die Oben-Unten-Ortung unterscheidet sich nur marginal voneinander.

In Abbildung 7.9 wird auf der Medianebene die Vorne-Hinten-Ortung von Sprachsignalen der von Musiksignalen gegenübergestellt. Beide Signale wurden in beiden Fällen von vorne wahrgenommen. Eine große Abweichung auf Grund des Signaltyps lässt sich nicht erkennen.

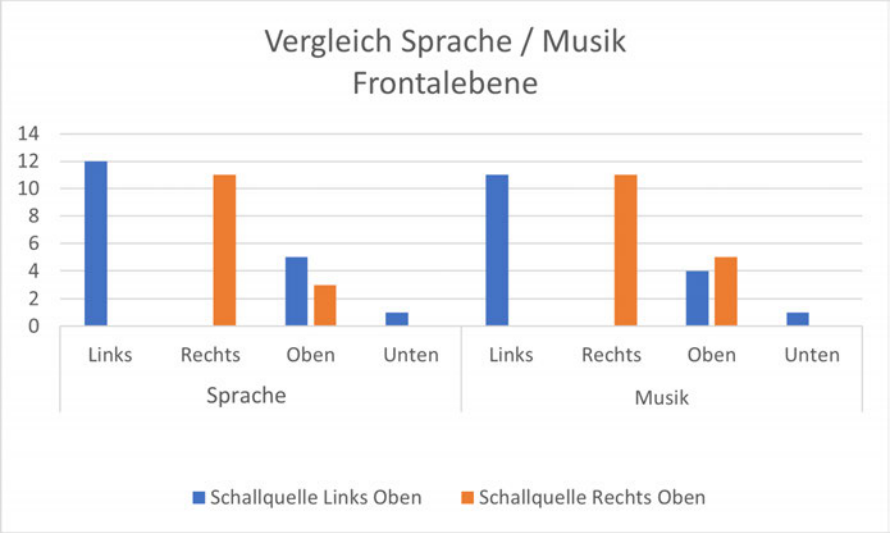

Abbildung 7.8: Vergleich von Sprache und Musik auf der Frontalebene (Eigene Darstellung)

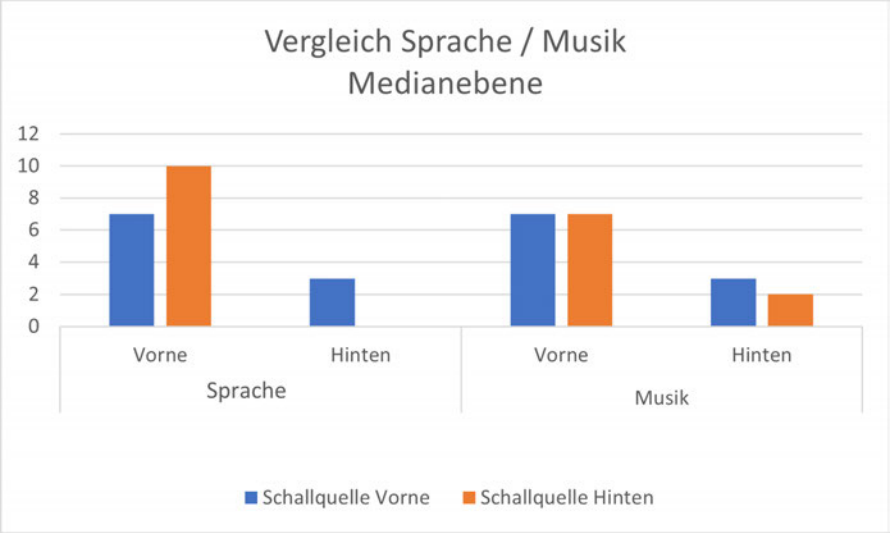

Abbildung 7.9: Vergleich von Sprache und Musik auf der Medianebene (Eigene Darstellung)

Interpretation

Für die Interpretation der Ergebnisse werden die theoretischen Grundlagen aus Kapitel 2 herangezogen. So lässt sich die fast fehlerfreie Ortung der linken und rechten Seite auf der Horizontalebene durch die Bestimmung von ILD und ITD erklären. Diese Laufzeit und Pegeldifferenzen sind bei der Schallortung des Menschen evolutionär besonders ausgeprägt. Die Ortung auf der Medianebene hingegen wird nur durch Interferenzeffekte der Ohrmuschel und des Oberkörpers möglich und ist bei jedem Menschen individuell ausgebildet.
Da bei der Verwendung einer HRTF-Datenbank wie der verwendeten KEMAR-Datenbank nur eine dem Durchschnitt angeglichene Ohrform berücksichtigt wird, ist die Ortung auf dieser Ebene bei jedem Menschen unterschiedlich gut ausgeprägt. Des Weiteren ist die Ortung in dieser Testreihe durch eine unbewegliche Punktschallquelle erschwert. Die im Kapitel 2 beschriebenen unterbewussten Kopfbewegungen zum besseren Bestimmen der Position sind hier nicht möglich und tragen zu einem schlechten Abschneiden der Positionierung auf der Medianebene bei.

Auffallend sind jedoch die überwiegend korrekten Ortungen von Schallquellen, die ober- bzw. unterhalb der hörenden Person positioniert sind. Auch hier sind für eine erfolgreiche Lokalisierung einzig die Interferenzeffekte der Ohrmuschel und des Oberkörpers verantwortlich. Auch die deutlich bessere Vorne-Hinten-Ortung bei Schallquellen, die zusätzlich noch aus der Medianebene nach links bzw. rechts verschoben wurden, ist auffällig. Während bei einer Positionierung in der Mitte bei $0°\varphi$ die hinten angeordnete Schallquelle keine einzige korrekte Zuordnung erfahren konnte (siehe Abbildung 7.4), wurde bei einer Verschiebung der Schallquelle nach rechts bzw. links fast zwei Drittel der Position auf der Medianebene richtig zugeordnet (siehe Abbildung 7.7).

8 Fazit

Ziel dieser Arbeit war die Entwicklung eines binauralen Positionierungsalgorithmus zur dynamischen Anordnung von Schallereignissen in einem virtuellen dreidimensionalen Raum. Im Rahmen dieses Kapitels sollen die erarbeiteten Erkenntnisse sowie ein Ausblick auf mögliche Forschungsfelder gegeben werden, die an diese Ausarbeitung anknüpfen.

8.1 Zusammenfassung

Um den Algorithmus zu entwerfen, wurde auf die Theorie der menschlichen Wahrnehmung von Schallereignissen in den unterschiedlichen Lokalisationsebenen und deren Auswirkung auf die Schallortung eingegangen. Dazu wurde die interaurale Lokalisation mit der Pegel- sowie der Laufzeitdifferenz der monoauralen Lokalisation mit ihren frequenzabhängigen Übertragungsfunktionen gegenübergestellt. Es wurde aufgezeigt, wie diese Ortungsinformationen mittels spezieller Messverfahren als Impulsantworten in HRTF-Dateien hinterlegt sind und in welcher Form diese Dateien zur Bearbeitung von Audiosignalen in Computersystemen eingesetzt werden. Diesbezüglich wurden die Grundlagen der digitalen Signalverarbeitung eingeführt und verschiedene Filtermethoden sowie deren Vor- und Nachteile diskutiert. Anschließend wurde ein Algorithmus zur Positionierung von Audiodateien in einem virtuellen dreidimensionalen Raum vorgestellt, bei dessen Entwicklung folgende Erkenntnisse gewonnen werden konnten:

1. HRIR-Daten können in unterschiedlichen Dateiformaten vorliegen und müssen separat eingelesen und verarbeitet werden können.

2. Die Positionsdaten der HRIR-Dateien können als kartesische Koordinaten oder als sphärische Koordinaten hinterlegt sein.

3. Die Faltung kann mittels eines FIR-Filters in der Zeitdomäne vorgenommen werden, was sich in einer simplen Implementierung widerspiegelt.

4. Um den Faltungsprozess effizient zu gestalten, muss das Audiosignal und die HRIR mittels FFT in die Frequenzdomäne überführt werden.

5. Die Konvertierung in die Frequenzdomäne mittels FFT birgt weitere Probleme, die mittels Implementierung spezieller Buffer gelöst werden können.

Mit Hilfe der gewonnen Erkenntnisse konnten die technischen Voraussetzungen sowie ein Anwendungsdesign für einen Algorithmus zum positionieren von Schallereignissen skizziert werden. Die Implementierung des Algorithmus wurde mittels des JUCE-Frameworks auf Basis der Programmiersprache C++ konzipiert und umgesetzt. Die durch den Algorithmus prozessierten Dateien wurden abschließend in einer Testreihe evaluiert und die Ergebnisse interpretiert.

8.2 Ausblick

Im Rahmen der Arbeit konnte gezeigt werden, dass die Entwicklung eines binauralen Positionierungsalgorithmus eine interdisziplinäre Zusammenarbeit zwischen der Informatik und dem physikalischen Teilgebiet der Schalllehre voraussetzt. Auf beiden Teilgebieten ergeben sich dabei Möglichkeiten für weiterführende Studien, um die Genauigkeit und Qualität des Algorithmus zu optimieren und zu verbessern. Eine naheliegende Methode, um eine höhere Auflösung der Positionierung zu erreichen, besteht in der Möglichkeit, die HRTF-Daten mit einer feineren Abstufung der Azimut- oder Elevationwinkel aufzunehmen. Auch auf der digitalen Ebene könnte durch eine Interpolation der einzelnen HRTF-Messpunkte ein realistischeres Verhalten bei einem dynamischen Wechsel der HRTF-Messpunkte erzielt werden.

Des Weiteren könnte die in dieser Arbeit am Rande beleuchtete Genauigkeit der Ortung in Bezug auf die Entfernung des Schallereignisses erhöht werden. Dies könnte durch eine Nutzung von HRTF-Datenbanken geschehen, die bei der Messung unterschiedliche Entfernungen im Nahbereich der Hörer:in berücksichtigen. Auch eine Verwendung von Rauminformationen durch künstlich generierten Nachhall stellt eine Möglichkeit dar, die Schallortung und das Realitätsempfinden zu steigern.

Einen interessanten Ansatz bietet auch eine Interaktionsschnittstelle mit dem System mittels eines sensorgesteuerten Wechsels der Impulsantworten. Zu diesem Zweck könnten die Bewegungssensoren eines Smartphones genutzt werden oder Algorithmen für bereits existierende VR-Systeme mit entsprechender Sensorik entwickelt werden.

Zusammenfassend lässt sich sagen, dass die Verwendung von binauraler Audiorepräsentation eine Vielzahl an Möglichkeiten in dem aufstrebenden Markt der Augmented- und Virtual-Reality-Anwendungen bietet, um weitere, verbesserte Algorithmen zur Positionierung von Schallereignissen in einem virtuellen Raum zu realisieren.

Literaturverzeichnis

[1] *Perception Lecture Notes: Auditory Pathways and Sound Localization*. September 2006. – URL https://www.cns.nyu.edu/~david/courses/perception/lecturenotes/localization/localization.html. – [Online; accessed 26. Apr. 2022]

[2] *Fachgebiet Audiokommunikation: Fachgebiet Audiokommunikation*. Juni 2022. – URL https://www.ak.tu-berlin.de/menue/fachgebiet_audiokommunikation. – [Online; accessed 13. Jun. 2022]

[3] *File:Overlap-add algorithm.svg - Wikimedia Commons*. Juni 2022. – URL https://commons.wikimedia.org/wiki/File:Overlap-add_algorithm.svg#/media/File:Overlap-add_algorithm.svg. – [Online; accessed 20. Jun. 2022]

[4] *SOFA (Spatially Oriented Format for Acoustics) - Sofaconventions*. Mai 2022. – URL https://www.sofaconventions.org/mediawiki/index.php/SOFA_(Spatially_Oriented_Format_for_Acoustics). – [Online; accessed 16. Jun. 2022]

[5] *The Brain and Space*. April 2022. – URL https://de.coursera.org/learn/human-brain. – [Online; accessed 28. Apr. 2022]

[6] *Wav file format - musicg-api*. Juni 2022. – URL https://sites.google.com/site/musicgapi/technical-documents/wav-file-format. – [Online; accessed 16. Jun. 2022]

[7] AL., Majdak et: Spatially Oriented Format for Acoustics: A Data Exchange Format Representing Head-Related Transfer Functions. In: *134th Audio Engineering Society Convention 2013* (2013), Mai. – URL https://www.researchgate.net/publication/236634182_Spatially_Oriented_Format_for_Acoustics_A_Data_Exchange_Format_Representing_Head-Related_Transfer_Functions

[8] Arar, Dr. Steve: Circular Buffer: A Critical Element of Digital Signal Processors. In: *All About Circuits* (2017), November. – URL https://www.allaboutcircuits.com/technical-articles/circular-buffer-a-critical-element-of-digital-signal-processors. – [Online; accessed 12. May 2022]

[9] Autoren der Wikimedia-Projekte: *C++ – Wikipedia*. Juli 2002. – URL https://de.wikipedia.org/w/index.php?title=C%2B%2B&oldid=223676626. – [Online; accessed 24. Jun. 2022]

[10] Autoren der Wikimedia-Projekte: *Blauertsche Bänder – Wikipedia*. Juli 2005. – URL https://de.wikipedia.org/w/index.php?title=Blauertsche_B%C3%A4nder&oldid=172310304. – [Online; accessed 27. Apr. 2022]

[11] Batteau D., W.: The role of the pinna in human localization. In: *Proc. R. Soc. Lond. B Biol. Sci.* 168 (1967), August, Nr. 1011, S. 158–180. – ISSN 2053-9193

[12] BennyOe: *binaural_conv*. August 2022. – URL https://github.com/BennyOe/binaural_conv. – [Online; accessed 1. Aug. 2022]

[13] BennyOe: *binaural_evaluation*. August 2022. – URL https://github.com/BennyOe/binaural_evaluation. – [Online; accessed 1. Aug. 2022]

[14] Blauert, Jens: *Spatial Hearing: The Psychophysics of Human Sound Localization*. Oxford, England, UK : Oxford University Press, Oct 1996

[15] Blauert, Jens: *The Technology of Binaural Listening*. Berlin, Germany : Springer, 2013. – ISBN 978-3-642-37761-7

[16] Contributors to Wikimedia projects: *JUCE - Wikipedia*. April 2022. – URL https://en.wikipedia.org/w/index.php?title=JUCE&oldid=1084109519. – [Online; accessed 22. Jun. 2022]

[17] Contributors to Wikimedia projects: *Real-time computing - Wikipedia*. März 2022. – URL https://en.wikipedia.org/w/index.php?title=Real-time_computing&oldid=1079337869. – [Online; accessed 21. Jun. 2022]

[18] Doumler, Timur: *C++ in the Audioindustry*. Juni 2022. – URL https://github.com/CppCon/CppCon2015/blob/master/Presentations/C++%20In%20the%20Audio%20Industry/C++%20In%20the%20Audio%20Industry%20-%20Timur%20Doumler%20-%20CppCon%202015.pdf. – [Online; accessed 21. Jun. 2022]

[19] Görne, Thomas: *Tontechnik: Schwingungen und Wellen, Hören, Schallwandler, Impulsantwort, Faltung, Sigma-Delta-Wandler, Stereo, Surround, WFS, Regiegeräte, tontechnische Praxis (Print-on-Demand)*. München, Germany : Carl Hanser Verlag GmbH & Co. KG, März 2011. – ISBN 978-3-44642395-4

[20] Gruenigen, Daniel von: *Digitale Signalverarbeitung: mit einer Einfuehrung in die kontinuierlichen Signale und Systeme*. Leipzig, Germany : Fachbuchverl. Leipzig im Carl-Hanser-Verlag, 2008. – ISBN 978-3-44641463-1

[21] Lee, Allen: *libBasicSOFA*. Juni 2022. – URL https://github.com/superkittens/libBasicSOFA. – [Online; accessed 16. Jun. 2022]

[22] Letowski, Tomasz ; Letowski, Szymon: Auditory Spatial Perception: Auditory Localization. In: *ResearchGate* (2012), Mai. – URL https://www.researchgate.net/publication/265033553_Auditory_Spatial_Perception_Auditory_Localization

[23] Litovsky, Ruth Y. ; Goupell, Matthew J. ; Fay, Richard R. ; Popper, Arthur N.: *Binaural Hearing*. Cham, Switzerland : Springer, 2021. – ISBN 978-3-030-57099-6

[24] Millot, Laurent ; Pelé, Gérard: An Alternative Approach for the Convolution in Time-Domain: The Taches-Algorithm. In: *ResearchGate* (2008), Mai. – URL https://www.researchgate.net/publication/277748792_An_Alternative_Approach_for_the_Convolution_in_Time-Domain_The_Taches-Algorithm/figures?lo=1

[25] Pirkle, W. C.: *Designing Audio Effect Plugins in C++*. Andover, England, UK : Taylor & Francis, 2019. – ISBN 978-0-42995432-0

[26] Sivonen, Ville ; Ellermeier, Wolfgang: Directional loudness in an anechoic sound field, head-related transfer functions, and binaural summation. In: *J. Acoust. Soc. Am.* 119 (2006), Juni, S. 2965–80

[27] Weinzierl, Stefan: *Handbuch der Audiotechnik*. Berlin, Germany : Springer, 2008. – ISBN 978-3-540-34300-4

[28] Wikimedia-Projekte, Autoren der: *Rainbow Books – Wikipedia*. Januar 2005. – URL https://de.wikipedia.org/w/index.php?title=Rainbow_Books&oldid=191867091. – [Online; accessed 6. May 2022]

[29] Xie, Bosun ; Zhang, Tingting: The Audibility of Spectral Detail of Head-Related Transfer Functions at High Frequency. In: *Acta Acustica United With Acustica - ACTA ACUST UNITED ACUST* 96 (2010), März, S. 328–339

[30] Zölzer, Udo: *DAFX: Digital Audio Effects*. Chichester, England, UK : Wiley, März 2011. – ISBN 978-0-47066599-2

Erklärung zur selbstständigen Bearbeitung

Hiermit versichere ich, dass ich die vorliegende Arbeit ohne fremde Hilfe selbständig verfasst und nur die angegebenen Hilfsmittel benutzt habe. Wörtlich oder dem Sinn nach aus anderen Werken entnommene Stellen sind unter Angabe der Quellen kenntlich gemacht.

Ort	Datum	Unterschrift im Original