Technische Universität Berlin
Fakultät I - Geisteswissenschaften
Institut für Sprache und Kommunikation
Fachgebiet Audiokommunikation

Masterarbeit

Entwicklung einer Versuchsumgebung zur Auralisation von virtuellen Konzerträumen für Musiker

Christoph Böhm

Technische Universität Berlin
Fakultät I - Geisteswissenschaften
Institut für Sprache und Kommunikation
Fachgebiet Audiokommunikation

Masterarbeit

Entwicklung einer Versuchsumgebung zur Auralisation von virtuellen Konzerträumen für Musiker

Vorgelegt von: Christoph Böhm
[redacted]

Erstgutachter: Prof. Dr. Stefan Weinzierl
Zweitgutachterin: Zora Schärer Kalkandjiev M.A.

Berlin, 10. September 2015

Erklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig und eigenhändig sowie ohne unerlaubte fremde Hilfe und ausschließlich unter Verwendung der aufgeführten Quellen und Hilfsmittel angefertigt habe.

Berlin, den 10. September 2015

Christoph Böhm

Danksagung

Mein besonderer Dank gilt Zora Schärer Kalkandjiev, die mich als Betreuerin mit ihrem Fachwissen und ihrer fortwährenden Unterstützung durch diese Arbeit geleitet hat. Bei Prof. Weinzierl bedanke ich mich für die Möglichkeit diese Arbeit zu schreiben. David Ackermann und Maximilian Ilse möchte ich für ihre umfangreiche Einführung in RAVEN und die Diskussionen danken, die so oft den richtigen Schalter im Kopf umgelegt und zur Lösung von Problemen geführt haben. Sönke Pelzer, Dirk Schröder und Michael Horn danke ich für ihre Hilfe mit RAVEN, dem Sound Scape Renderer und Linux. Den immer hilfsbereiten Mitarbeitern des Fachgebiets Audiokommunikation danke ich für ihre Unterstützung bei meiner Arbeit und gemeinsam mit Prof. Weinzierl für die spannende, abwechslungsreiche und zielführende Gestaltung des Masterstudiengangs Audiokommunikation.

Als wichtigste Unterstützer während meiner Arbeit und meiner gesamten Ausbildung sind jedoch meine Frau Jana, meine Eltern und Großeltern zu nennen. Sie haben mich immer wieder ermutigt, gestützt, angetrieben und mein Studium überhaupt erst möglich gemacht.

Zusammenfassung

In der Raumakustikplanung gewinnen computergestützte Simulationen zunehmend an Bedeutung. Durch die rasante Entwicklung der Rechengeschwindigkeit moderner Computer und die Verbesserung der Simulationsmethoden steigt die Leistungsfähigkeit solcher Werkzeuge weiter an. Es können Modelle von Räumen nahezu beliebiger Art und Größe berechnet werden. Zusätzlich zur Berechnung raumakustischer Parameter ist die Auralisation der simulierten Schallfelder heute unter anderem dazu geeignet, virtuelle akustische Umgebungen zu schaffen und die Simulationen akustisch erlebbar zu machen.

Im Kontext des Forschungsprojekts „Raumakustik und musikalische Interpretation“ am Fachgebiet Audiokommunikation wurde ein technisches Verfahren zur Erstellung von virtuellen Konzerträumen für Musiker mithilfe von dynamischer Binauralsynthese entwickelt. Ein Ensemble von vier Musikern kann in virtuellen akustischen Umgebungen miteinander musizieren. Dadurch ergibt sich die Möglichkeit, vielfältige Untersuchungen zur Wahrnehmung von Raumakustik durch Musiker und zum raumakustischen Einfluss auf die musikalische Interpretation durchzuführen.

Es wurden Computersimulationen von 14 Aufführungsstätten mit der Simulations- und Auralisations-Software RAVEN durchgeführt. Dabei wurde auf eine realitätsnahe Implementierung der Schallquellenpositionen und Richtcharakteristiken geachtet. In der Simulation wurden binaurale Raumimpulsantworten (BRIR) für zuvor definierte Freiheitsgrade der Empfänger gewonnen. Mit einem technischen Aufbau im reflexionsarmen Raum wurden damit virtuelle akustische Umgebungen für Musiker geschaffen. Diese wurden durch die Faltung der Quellsignale der vier Instrumente mit den berechneten BRIRs und die Wiedergabe über extra-aurale Kopfhörer mittels dynamischer Binauralsynthese realisiert. Alle computergestützten Berechnungen wurden mithilfe der Programmierung von Skripten automatisiert, um den späteren Einsatz der Umgebung in Hörversuchen zu erleichtern. In einem Vorversuch wurde anschließend die Funktionalität der Versuchsumgebung und die Plausibilität der Auralisation evaluiert.

Es konnte eine Versuchsumgebung entwickelt werden, die eine technisch korrekte und realitätsnahe Auralisation von virtuellen Konzerträumen ermöglicht. In der Evaluation des Versuchsaufbaus wurde eine hohe Immersion der Musiker in die akustische Umgebung festgestellt. Durch die Beachtung aller relevanten Einflüsse auf das virtuelle Schallfeld in der Simulation und die Echtzeit-Fähigkeit der Auralisation, eignet sich das System damit für den Einsatz als Versuchsumgebung in zukünftigen Studien des Forschungsprojekts.

STICHWORTE: Auralisation, virtuelle akustische Umgebung (VAE), dynamische Binauralsynthese, akustische Simulation, RAVEN, geometrische Akustik, binaurale Raumimpulsantwort (BRIR)

Abstract

These days computer-aided simulations become more and more important in the design process of room acoustics. Rapid improvements of computation speed and modern methods in computer simulation improve the quality and efficiency of those technologies. Room models of almost any size can be designed and simulated. Besides the calculation of room acoustic parameters the auralization of simulated sound fields offers the possibility to create virtual acoustic environments.

During the research project „Room Acoustics and the Performance of Music“ conducted by the Audio Communication Group a technical procedure was developed. Its purpose was to generate virtual concert rooms for musicians by means of dynamic binaural synthesis. In a virtual acoustic environment an ensemble of four musicians is enabled to play music. In this way miscellaneous investigations on the perception of room acoustics and the performance by musicians in varying acoustical situations can be done.

Computer simulations of 14 typical performance venues were conducted by using the acoustic simulation and auralization software RAVEN. Realistic source and receiver positions and characteristics were implemented to achieve authentic results. Binaural room impulse responses were calculated for previously defined degrees of freedom of the possible head orientations. These were used to create virtual acoustical environments with a technical setup in an anechoic chamber. The source signals of the four used instruments were convolved in real-time with the previously generated BRIRs and were played back by dynamic binaural synthesis via extra-aural headphones. All computer-aided processes were made automatic by Matlab- and bash-scripting to make the later use of the test environment more comfortable. In a subsequent pilot test the functionality and plausibility of the auralization were examined.

An experimental setup was developed that allows technically correct auralizations of virtual concert halls close to reality. During the pilot test a high degree of immersion of the musicians into the virtual acoustic environment could be observed. Due to the real-time capability and the proper implementation of all elements influencing the computer simulated sound field, the system is capable to be used in future studies of the Audio Communication Department.

KEYWORDS: auralization, virtual acoustic environment (VAE), dynamic binaural synthesis, acoustics simulation, RAVEN, geometrical acoustics, binaural room impulse response (BRIR)

Inhaltsverzeichnis

Abkürzungsverzeichnis **XVII**

1. Einleitung **1**

2. Auralisation von virtuellen akustischen Umgebungen **5**
2.1. Virtuelle Akustische Umgebungen 5
2.2. Binauraltechnik 7
2.2.1. Raumimpulsantwort (RIR) 8
2.2.2. Binaurale Messtechnik 10
2.2.3. Binauralsynthese 11
2.3. Akustische Simulation 13
2.3.1. Deterministische Spiegelschallquellenmethode 14
2.3.2. Stochastisches Ray Tracing 16
2.3.3. Hybride Methode 19

3. Methode **21**
3.1. Konzeption der virtuellen akustischen Umgebungen 22
3.1.1. Simulation mit RAVEN 24
3.1.2. Koordinatensysteme 24
3.1.3. Modelle 25
3.1.4. Positionierung der Musiker 30
3.1.5. Richtcharakteristiken 31
3.1.6. Freiheitsgrade 32
3.1.7. BRIR Berechnung 34
3.1.8. RAVEN Parameter 42
3.2. Versuchsaufbau 45
3.2.1. Technischer Aufbau 46
3.2.2. Aufnahme 49
3.2.3. Signalverarbeitung 50
3.2.4. Wiedergabe 54
3.2.5. Headtracker 56
3.2.6. Kalibrierung 56
3.2.7. Systemlatenz 59

4. Evaluation der Versuchsumgebung im Vorversuch **63**
4.1. Durchführung 63
4.2. Ergebnis des Vorversuchs 65

5. Zusammenfassung und Diskussion **67**

Literaturverzeichnis **69**

Abbildungsverzeichnis **73**

Tabellenverzeichnis **75**

A. Ausschnitte Matlab Code **77**

B. VSE Konfigurationsdateien **81**

C. VSE Ordnerstrukturen **83**

D. Plots und Bildschirmfotos **87**

E. Hardware-Patchplan **89**

F. Fotos **91**

G. Datenträger **95**

Abkürzungsverzeichnis

ASDF	Audio Scene Description Format
BSP	Binary Space Partitioning
BRIR	Binaural Room Impulse Response
BRS	Binaural Room Synthesis
BF	Brute Force
DRIR	Directional Room Impulse Response
DUT	Device Under Test
EASE	Electro-Acoustic Simulator for Engineers
FABIAN	Fast and Automatic Binaural Impulse response AcquisitioN
FEC	Free-air Equivalent Coupling
FEM	Finite Elemente Methode
GUI	Graphical User Interface
HATO	Head Above Torso Orientation
HATS	Head and Torso Simulator
HOA	Higher Order Ambisonics
HpTF	Headphone Transfer Function
HRIR	Head Related Impuse Response
HRTF	Head Related Transfer Function
JND	Just Noticeable Difference
openDAFF	open Directional Audio File Format
RAVEN	Room Acoustics for Virtual ENvironments
RAR	reflexionsarmer Raum
RIR	Room Impulse Response
RTF	Room Transfer Function
SH	Spatial Hashing
SOFA	Spatially Oriented Format for Acoustics

SSR	Sound Scape Renderer
VAE	Virtual Acoustic Environment
VSE	Virtual Stage Environment

1. Einleitung

Die Akustik spielt für alle Nutzungsarten von Räumen eine entscheidende Rolle für die Rezeption von Schallereignissen. Besondere Relevanz kommt der Akustik in musikalischen Aufführungsräumen zu. Die Raumakustik ist maßgeblich für die Eigenschaften der Übertragung des Schalls vom Sender zum Empfänger verantwortlich. Dabei bilden die Musiker mit ihren Instrumenten die Schallquellen, die den Raum anregen. Als Empfänger fungieren neben dem Publikum allerdings auch die Musiker selbst. Für sie ergibt sich daraus auch eine Abhängigkeit vom Einfluss der Raumakustik auf ihre Spielweise. In einem Wechselspiel regen sie selbst das Schallfeld im Raum an und reagieren auf die akustische Antwort des Raums.

Am Fachgebiet Audiokommunikation der TU Berlin werden Untersuchungen dieser Zusammenhänge im DFG-Forschungsprojekt „Raumakustik und musikalische Interpretation“ durchgeführt. Frühere Studien von Von Békésy (1968), Bolzinger et al. (1994), Kato et al. (2007, 2008) und Ueno et al. (2010) hatten sich bereits mit diesem Einfluss der Raumakustik auf die musikalische Interpretation beschäftigt. Sie registrierten dabei teilweise widersprüchliche Abhängigkeiten der Dynamik und des Tempos der Interpretation von einzelnen raumakustischen Parametern, wie der Nachhallzeit. Um eindeutigere Zusammenhänge feststellen zu können, ermittelten Schärer Kalkandjiev und Weinzierl (2013) im Rahmen des Forschungsprojekts deshalb zunächst für die Perzeption relevante Kriterien zur Charakterisierung der musikalischen Darbietung und setzten einen umfangreicheren Satz Parameter für die Raumakustik ein. Es wurde die Interpretation mehrerer Musikstücke durch einen Musiker in sieben realen Räumen beobachtet. Die Darbietungen wurden aufgezeichnet und mithilfe computergestützter Signalverarbeitung untersucht. Außerdem wurden raumakustische Parameter gemessen. In der statistischen Auswertung der Ergebnisse konnten klare Bezüge zwischen verschiedenen Parametern der Interpretation und Raumakustik festgestellt werden.

Um die gewonnenen Erkenntnisse zu unterstreichen wurde durch Schärer Kalkandjiev und Weinzierl (2015) eine weitere Studie durchgeführt. Die wichtigste Veränderung bestand in der Verlagerung des Versuchs in eine Laborumgebung. So wurden nunmehr 14 verschiedene computersimulierte Räume mittels dynamischer Binauralsynthese für die Musiker auralisiert. Mit der Durchführung des Experiments in einem Labor ergibt sich eine Lösung von äußeren, nicht akustischen Einflüssen der Konzerträume sowie die Möglichkeit an einem Ort beliebig viele akustische Umgebungen für die Versuche zu verwenden. Außerdem wurde es möglich, mehrere Musiker in die Versuche einzubinden und zwei Musikstücke unterschiedlicher Tempi zu untersuchen.

Eine Beziehung zwischen Nachhallzeit und Tempo konnte in beiden Studien durch Schärer Kalkandjiev und Weinzierl (2013, 2015) nachgewiesen werden. Allerdings stellte sich heraus, dass die Musiker individuell verschieden auf unterschiedliche akustische Umgebungen reagierten. So fand zwar grundsätzlich eine Anpassung des Tempos an die Umgebung statt, sie geschah jedoch auf unterschiedliche Weise. Die Dynamik und die Klangfarbe stellten sich dagegen, bei unterschiedlichen Musikern und Musikstücken, als homogener abhängig von raumakustischen Parametern dar.

Nach diesen gewonnen Erkenntnissen stellt sich nun die Frage, wie das Verhalten mehrerer Musiker im Ensemble durch verschiedene akustische Umgebungen beeinflusst wird. Es ist zu erwarten, dass sie auch ihr Zusammenspiel an die Raumakustik anpassen, da der gemeinsame Klang ihrer Instrumente durch die akustische Antwort des Raums bestimmt wird. Jedoch ist zu klären, ob dieselben raumakustischen Parameter maßgeblich für die Anpassung ihrer musikalischen Interpretation verantwortlich sind oder ob neue Faktoren eine Rolle spielen.

Diese Masterarbeit hat zum Ziel, eine geeignete Versuchsumgebung zur Bearbeitung dieser Fragestellung zu schaffen. Ein Streichquartett soll in verschiedenen virtuellen Räumen gemeinsam musizieren können. Die Entwicklung der Umgebung teilt sich in zwei Kernbereiche auf: Zunächst müssen die Schallfelder der virtuellen Räume simuliert und die zugehörigen binauralen Impulsantworten (BRIR) berechnet werden. Hierfür soll die Raumakustik-Simulations-Software RAVEN[1] zum Einsatz kommen. Diese Berechnung soll mithilfe eines Matlab-Skripts automatisiert werden, sodass eine beliebige Anzahl von Räumen mit mehreren Quellen und Empfängern innerhalb einer Berechnung simuliert werden kann. Für die Simulation mit RAVEN müssen die Modelle von Schärer Kalkandjiev und Weinzierl (2015) in SketchUp-Modelle überführt werden. Außerdem sind die passenden Richtcharakteristiken der Instrumente und binauralen Empfänger sowie deren korrekte Positionen innerhalb eines Ensembles und einer Aufführungsstätte zu implementieren.

Für das gemeinsame Musizieren der Musiker in den virtuellen akustischen Umgebungen wird ein technischer Aufbau im reflexionsarmen Raum als Laborumgebung entwickelt. Die Auralisation des Schallfeldes für jeden Musiker wird mit dynamischer Binauralsynthese umgesetzt. Die Quellsignale werden direkt am Instrument aufgenommen und nach einer Faltung in Echtzeit mit den zuvor berechneten binauralen Raumimpulsantworten über geeignete Kopfhörer für jeden Musiker wiedergegeben. Die Auralisation dient hier nur der Wiedergabe des diffusen Schallfeldes und der Reflexionen, da der Direktschall die Musiker im RAR[2] direkt von ihren Instrumenten erreicht. Um das richtige Verhältnis zwischen dem Direktschall und dem Diffusschall und den Reflexionen im synthetisierten Schallfeld zu erreichen, ist eine Methode zur Kalibrierung dieses Verhältnisses zu entwickeln. Die Versuchsumgebung soll

[1] Room Acoustics for Virtual ENvironments
[2] reflexionsarmer Raum

durch eine geeignete Menüführung und Softwarestruktur den Versuchsleiter in der Durchführung der Kalibrierung und des eigentlichen Versuchs unterstützen.

Zur Evaluation der Versuchsumgebung wird ein Vorversuch durchgeführt. Hier werden zum einen der technische Aufbau und die entwickelten Methoden zur Kalibrierung des Wiedergabesystems auf ihre Funktionalität überprüft, zum anderen wird die Plausibilität der Auralisation durch Interviews mit den teilnehmenden Musikern untersucht.

2. Auralisation von virtuellen akustischen Umgebungen

Wie auch in anderen Ingenieurwissenschaften werden in der Akustik physikalische Probleme oder Eigenschaften durch messbare Parameter beschrieben. Auf Grundlage der zeitlichen und spektralen Struktur eines Signals werden Größen, wie Nachhallzeit, Stärkemaß oder Schärfe und Klarheit, definiert. Neben diesen Bewertungskriterien ist es jedoch auch möglich und oftmals auch notwendig, akustische Eigenschaften hörbar zu machen. Gerade in der Entwicklung und Forschung werden durch Messungen die Eigenschaften von akustischen Systemen erfasst oder, durch computergestützte Simulationen, Modelle von akustischen Systemen entwickelt. Hierbei werden Daten gewonnen, welche hörbar gemacht werden können. Dieser Vorgang der „Hörbarmachung“ von numerisch simulierten, gemessenen oder synthetisierten Daten wird Auralisation genannt (Vorländer, 2007, S. 3).

Für Entwicklungen im Schallschutz können Verkehrsgeräusche und Fluglärm simuliert und Schutzmaßnahmen gegen den Lärm entwickelt werden. In der Konstruktion von neuen Bauteilen oder Maschinen ist es möglich, deren akustische Wirkung bereits in der Planungsphase zu auralisieren und zu beurteilen. Im Planungsprozess von Räumen verschiedener Nutzungsart, wie Konzerthäusern, Konferenzräumen, Klassenzimmern und Großraumbüros, kann die Akustik schon vor dem Bau oder Umbau detailliert analysiert und hörbar gemacht werden. Mit entsprechenden Computermodellen dieser Räume und der Simulation ihrer Schallfelder, können ganze virtuelle akustische Umgebungen geschaffen werden, in die ein Nutzer eintauchen und den Klang einer virtuellen Schallquelle erleben kann. Für die Wissenschaft hat sich diese Technik als probates Mittel erwiesen, um ohne Ortsbindung Hörversuche in verschiedenen akustischen Umgebungen durchzuführen. Die Auralisation der virtuellen Schallfelder kann dabei mit verschiedenen Methoden durchgeführt werden, von denen in dieser Arbeit die dynamische Binauralsynthese zum Einsatz kommen soll.

2.1. Virtuelle Akustische Umgebungen

Im hier betrachteten Fall der Raumakustiksimulation und der zugehörigen Auralisation entsteht häufig die Forderung nach der Möglichkeit einer Interaktion des Benutzers mit der akustischen Umgebung. Die Technologie der virtuellen Realität greift diese Forderung auf. Sie wird von Burdea und Coiffet (1994, S. 3) als Benutzerschnittstelle definiert, die auf der Wahrnehmung von und Interaktion mit visuellen, akustischen, haptischen, gustatorischen und olfaktorischen Reizen basiert.

Durch computergenerierte Objekte werden virtuelle Umgebungen geschaffen, in die der Benutzer eintauchen und mit denen er in Echtzeit interagieren kann. Zu einer virtuellen Realität gehören laut Burdea und Coiffet immer die drei Faktoren Immersion, Interaktion und Imagination. Der Benutzer soll in die virtuelle Umgebung eingebunden werden und seine reale Umgebung vergessen. Durch Erfassung seiner Blickrichtung, Position und Aktivitäten durch das System, interagiert der Benutzer mit der virtuellen Umgebung und hat die Möglichkeit diese zu beeinflussen. Das erlaubt ihm die Immersion in die Umgebung und vermittelt ihm den Eindruck einer realen Existenz der Objekte.

In diesem Zusammenhang der Entwicklung von virtuellen Umgebungen spielt die Akustik eine immer größere Rolle. Zusätzlich zu den, durch den Benutzer in einer Interaktion mit seiner Umgebung hervorgerufenen Schallereignissen, soll die Akustik von virtuellen Räumen dem visuellen Eindruck des Nutzers entsprechen. Durch die Erweiterung der Umgebung um kohärente Stimuli wird die Immersion des Nutzers in der virtuellen Realität gesteigert (Schröder, 2011, S. 4). Für bestimmte Anwendungsfelder ist eine virtuelle Realität mit der komplexen Gesamtheit ihrer Reize jedoch nicht notwendig oder sogar nicht erwünscht. Zum Beispiel können in einer rein akustischen virtuellen Umgebung Reaktionen des Nutzers auf die Akustik, losgelöst von visuellen Reizen, untersucht werden.

Im Rahmen dieser Arbeit bietet der Einsatz einer virtuellen akustischen Umgebung (VAE[1]) die Möglichkeit verschiedene Räume für zusammen musizierende Musiker in Echtzeit zu auralisieren. Durch die Nutzung einer virtuellen Umgebung, im Gegensatz zu realen Räumen, kann nahezu nahtlos zwischen den verschiedenen Räumen gewechselt werden. Dadurch wird ein direkter Vergleich zwischen den akustischen Eigenschaften und ihrem Einfluss auf die Musiker möglich. Bei einer solchen Anwendung spielen akustische Eigenschaften des Raums, die durch die Geometrie beeinflusst werden und vor allem die richtige Ortung der Schallquellen eine Rolle. So muss das künstlich erzeugte Schallfeld an die Position und Blickrichtung des Musikers angepasst werden. Zu diesem Zweck wird in einer Computersimulation ein Modell des zu untersuchenden Raums erstellt. In diesem Modell werden Quellen und Empfänger eingebunden und möglichst naturgemäß nachgebildet. Etwaige Freiheitsgrade in der Bewegung des Nutzers werden definiert und ihre Auswirkungen auf das Schallfeld berechnet. Die zur Positionsänderung passenden Daten können später in der Auralisation dynamisch geladen werden. Dabei ist zu beachten, dass, trotz der rasanten Entwicklung moderner Computer, die Qualität einer Auralisation immer abhängig von der Rechenleistung und Qualität der Simulation ist. Die Berechnungen und die Auralisation müssen jedoch nicht physikalisch perfekt, sondern für die menschliche Wahrnehmung korrekt sein (Vorländer, 2007, S. 5). Außerdem haben Moldrzyk et al. (2005) festgestellt, dass ohne unmittelbaren Vergleich zur realen akustischen Umgebung, eine qualitativ hochwertige, binaural simulierte Umgebung, von einer Mehrzahl an Versuchsteilnehmern, nicht von einer realen

[1] Virtual Acoustic Environment

Umgebung unterschieden werden kann. Lindau und Weinzierl (2012) verwenden hierfür den Begriff der Plausibilität als Bewertung für VAEs, verbunden mit der Frage, ob eine Simulation eines akustischen Ereignisses mit den Erwartungen des Hörers übereinstimmt. Mit optimierten Algorithmen zur Simulation von veränderlichen Schallfeldern und paralleler Berechnung über mehrere Computer ist heute in Grenzen auch die Berechnung und gleichzeitige Auralisation von VAEs in Echtzeit möglich (Schröder, 2011, S. 84). Um eine hohe Qualität der Simulation zu erreichen und auch hochauflösende Datensätze ohne Einschränkung mit niedigen Latenzen nutzen zu können, ist jedoch eine Berechnung der Umgebungen im Voraus notwendig. Das in dieser Arbeit beschriebene Verfahren zur Erstellung einer VAE wird deshalb in zwei Teile unterteilt: Die Computersimulation der virtuellen akustischen Umgebung anhand eines Modells und die Interaktion des Nutzers mit dieser vorher berechneten Umgebung, auralisiert mit einem technischen Aufbau im Labor.

Für die Auralisation von VAEs werden verschiedene Technologien eingesetzt. Grundsätzlich kann zwischen der Audiowiedergabe über Kopfhörer oder über Lautsprecher unterschieden werden. Für die Wiedergabe über Lautsprecher kommen die binaurale Wiedergabe über zwei Lautsprecher für einen Hörer oder Mehrkanalsysteme mit Technologien wie Higher Order Ambisonics (HOA) und Wellenfeldsynthese zum Einsatz. Letztere Methoden können auch mit mehreren Nutzern verwendet werden. Allerdings können laut Lindau et al. (2007) nicht alle räumlichen Eigenschaften des Schallfelds mit diesen Methoden synthetisiert werden. Außerdem haben physikalische Artefakte wie räumliches Aliasing Einfluss auf die Genauigkeit der Auralisation. Der zudem hohe technische Aufwand solche Systeme zu betreiben kann, durch die für jeden Nutzer getrennt durchgeführte Synthese der einzelnen Ohrsignale mithilfe von Kopfhörern, verringert werden. Solange die Anzahl der Nutzer nicht zu hoch wird und das Tragen von Kopfhörern das immersive Erleben nicht zu stark beeinflusst, bietet sich diese Methode an und ermöglicht eine lebensnahe, realistische Reproduktionen von beliebigen Schallfeldern (Lindau, 2014; Lindau und Weinzierl, 2012). Auch im Versuchsaufbau dieser Arbeit soll die VAE über Kopfhörer realisiert werden. Die zugrunde liegende Methode wird Binauraltechnik genannt und soll im folgenden Kapitel erläutert werden.

2.2. Binauraltechnik

Die binaurale Aufnahme- und Wiedergabetechnik bietet, gegenüber anderen mehrkanaligen oder monauralen Techniken, die Möglichkeit ein Schallfeld für einen Hörer ortsunabhängig mit allen seinen akustischen Eigenschaften zu reproduzieren. Wie von Møller (1992, S. 171,172) beschrieben, kann die gesamte auditive Wahrnehmung, die ein Hörer am Ursprungsort eines Schallereignisses erfährt, festgehalten und dichotisch[2] über Kopfhörer oder Lautsprecher wiedergeben werden. Sowohl die

[2]Beide Ohren werden unterschiedlich beschallt.

Klangfarbe als auch die richtungsabhängigen Eigenschaften des Schallfeldes werden dabei berücksichtigt und rekonstruiert. Lindau (2014) konnte zeigen, dass die Binauraltechnik ein geeignetes Mittel zur Durchführung von Hörversuchen darstellt und die auralisierten Umgebungen von Probanden, trotz möglicher Abweichungen von realen Schallfeldern, als plausibel angesehen werden (Lindau und Weinzierl, 2012). Dabei ist es sowohl möglich ein spezifisches Schallereignis, wie zum Beispiel ein Konzert aufzunehmen, als auch messtechnisch die Eigenschaften des Raums unabhängig von der Quelle zu ermitteln oder diese in einer Computersimulation zu berechnen. Hierbei wird eine Raumimpulsantwort (RIR) zur Beschreibung des als System betrachteten Raums ermittelt. In einer solchen Messung oder Berechnung sind die Position der Quelle und des Empfängers im Raum festgeschrieben. Wird die RIR monaural bestimmt, ist keine Richtungsinformation in der Raumantwort enthalten. Die Binauraltechnik macht sich die Eigenschaft des menschlichen Gehörs zunutze, das aus zwei getrennten Ohr-Signalen ein Signal mit räumlicher Information der aufgenommenen Schallereignisse extrahiert. Mit Kunstkopfmikrofonen kann eine binaurale Raumimpulsantwort (BRIR) gewonnen werden, die diese räumlichen Informationen ebenfalls enthält. In der Binauralsynthese kann diese gemessene BRIR mit einem unabhängigen nachhallfreien Signal gefaltet und über ein geeignetes binaurales Wiedergabesystem zur Rekonstruktion des ursprünglichen Schalfelds wiedergegeben werden.

Um die grundlegende Methode zu beleuchten, soll hier zunächst auf die allgemeine Methode der binauralen Aufnahme und Wiedergabe sowie die Binauralsynthese eingegangen werden. Die computergestützte Simulation von akustischen Umgebungen, welche anstelle von binauralen Aufnahmen oder Messungen treten kann, um virtuelle akustische Umgebungen zu erzeugen, wird dann in Abschnitt 2.3 betrachtet.

2.2.1. Raumimpulsantwort (RIR)

Die Messtechnik und Berechnungsmethoden für RIRs basieren auf der Grundannahme, dass sich ein Raum mit einem Schallübertragungsweg von einer Quelle zum Empfänger wie ein weitgehend lineares, zeitinvariantes System verhält. Ein solches System kann durch seine Impulsantwort $h(t)$ vollständig beschrieben werden (Ahnert und Tennhardt, 2008, S. 286). Wird ein Raum mit einem Signal $s(t)$ angeregt, gilt für das Ausgangssignal $g(t)$ an der Empfängerposition im Raum:

$$g(t) = h(t) * s(t) \tag{2.1}$$

Mithilfe des Faltungssatzes, wird aus der Faltungsoperation im Zeitbereich eine Multiplikation im Frequenzbereich (Möser, 2012, S.436):

$$G(f) = H(f) \cdot S(f) \tag{2.2}$$

Durch Division durch das Eingangssignal kann die Übertragungsfunktion $H(f)$ des Raums (RTF[3]) berechnet werden:

$$H(f) = \frac{G(f)}{S(f)} \tag{2.3}$$

Die inverse Fouriertransformation ergibt die gesuchte RIR $h(t)$:

$$h(t) = \mathcal{F}^{-1}\{H(f)\} \tag{2.4}$$

Die Übertragungsfunktion an der Position eines Empfängers im Schallfeld setzt sich aus mehreren Teilen zusammen. Zur Erläuterung dieser Zusammensetzung muss der gesamte Signalpfad betrachtet werden, den ein Signal von der Quelle bis zum Empfänger durchläuft. Die verschiedenen Übertragungspfade, die der Schall in alle Raumrichtungen verteilt durchlaufen kann, können als einzelne LTI-Systeme aufgefasst werden. Die Summe aller Systeme ergibt das Gesamtsystem des betrachteten Raums.

Ausgesendet von einer Quelle, die durch ihre spezifische Richtcharakteristik eine Übertragungsfunktion H_D aufweist, wird eine Schallwelle im Raum an den Begrenzungsflächen des Raums reflektiert, absorbiert oder gebrochen. Dadurch verliert die Welle an Energie. Dieser Verlust ist frequenzabhängig und von der Beschaffenheit der getroffenen Oberfläche abhängig. Jedes Zusammentreffen mit einer Oberfläche kann also durch eine weitere Übertragungsfunktion H_E beschrieben werden. Auf dem Weg durch den Raum kommt die Beeinflussung des Schalls durch die Luftabsorption H_{Air} hinzu. Trifft die Schallwelle zum Schluss auf den Empfänger müssen seine akustischen Eigenschaften, die aus seiner Geometrie und Oberflächenbeschaffenheit resultieren, durch seine Übertragungsfunktion H_{Head} mit in das System aufgenommen werden. Die Übertragungsfunktion des gesamten LTI-Systems ergibt sich also nach Schröder (2011, S. 23) aus der Multiplikation der Teilsysteme:

$$H_{Reflection} = H_D \cdot H_{Air} \cdot H_{Head} \cdot \prod_{HitWalls} H_{E_i} \tag{2.5}$$

$H_{Reflection}$ beschreibt also die Übertragungsfunktion eines Übertragungsweges der Schallwelle durch den Raum. Soll ein Raum und sein akustisches System durch eine Impulsantwort beschrieben werden, wird ein ausgesendeter Impuls auf jedem seiner Übertragungswege mit der Impulsantwort $h_{Reflection}$ der Teilsysteme gefaltet. Die Summe aller Impulsantworten am Empfänger ergibt die Raumimpulsantwort h. Sie beschreibt den Raum mit seinen akustischen Eigenschaften an der Empfängerposition vollständig. Diese in ihrer einfachsten Form monaural mit einem Mikrofon

[3]Room Transfer Function

bestimmte Impulsantwort kann sowohl für messtechnische Zwecke als auch für die hier betrachtete Auralisation binaural mit zwei Mikrofonen und einem geeigneten Messaufbau als binaurale Raumimpulsantwort (BRIR) gemessen werden.

2.2.2. Binaurale Messtechnik

Die Messung von BRIRs kann mit Miniaturmikrofonen in den Ohren eines Hörers oder unter Verwendung eines Kunstkopfes mit integrierten Mikrofonen durchgeführt werden. In einem Raum, dessen Impulsantwort bestimmt werden soll, wird der Kunstkopf auf einer gewünschten Empfängerposition aufgestellt. Diese Position im Raum wird fest in die RIR kodiert. Als Quelle dient ein Lautsprecher, dessen ausgesendetes Messsignal über die Kunstkopfmikrofone aufgezeichnet wird. Die Entfaltung der Raumimpulsantwort findet, entsprechend Gleichung 2.3, allerdings für beide Mikrofonsignale getrennt statt.

Die meisten Kunstkopfmodelle bilden die durchschnittliche Physiognomie eines menschlichen Kopfes nach. Dazu gehören die Augenhöhlen, die Nase sowie die Ohrmuschel und der Ohrkanal (Møller, 1992, S. 172). Wie Untersuchungen von Algazi et al. (2002) gezeigt haben, ist auch der Torso von Bedeutung für die Übertragungsfunktion des Ohrs und wird deswegen bei neueren Entwicklungen, wie dem Kunstkopf-Torso-Simulator FABIAN[4], mit einbezogen (Lindau und Weinzierl, 2007). Für die Anordnung der beiden benötigten Mikrofone im Kunstkopf kommen verschiedene Varianten in Frage. Das FABIAN-System verfügt über bündig im Boden der Ohrmuschel eingelassene Miniaturkondensatormikrofone, die vom Körperschall entkoppelt sind.

Durch Reflexion, Beugung und Abschattung nehmen die äußeren Körperteile Einfluss auf die in Abbildung 2.1 dargestellte Außenohrübertragungsfunktion (HRTF[5]). Diese, bei jedem Menschen individuelle Übertragungsfunktion, bestimmt das räumliche Hören. Durch Unterschiede zwischen dem linken und rechten Ohr kann im Gehirn ein Richtungsbezug für einfallenden Schall hergestellt werden. Physikalisch kann dies durch den Unterschied zwischen den Übertragungsfunktionen der beiden Ohren begründet werden. Durch die Division beider Funktionen wird die interaurale Außenohrübertragungsfunktion gebildet. Diese enthält Informationen über Unterschiede zwischen linkem und rechtem Ohr, die in der Färbung, der Phase, dem Zeitversatz und dem Pegel liegen können. Diese Größen sind nach Møller (1992, S. 176) verantwortlich für die Lokalisation der Schallquelle in allen Raumrichtungen.

Korpus, Schulter, Kopf und Ohrmuschel bilden Faktoren, die von der Schalleinfallsrichtung abhängig sind. Im Gegensatz dazu sind Einflüsse des Ohrkanals und des Ohreingangs (cavum conchae) unabhängig von der Schalleinfallsrichtung. Eben diese Einflüsse haben jedoch laut Weinzierl (2008, S. 586) oberhalb von 1kHz große

[4]Fast and Automatic Binaural Impulse response AcquisitioN
[5]Head Related Transfer Function

Wirkung auf den Frequenzgang und müssen für eine universelle Anwendbarkeit einer gemessenen BRIR kompensiert werden. Aufgrund der Richtungsunabhängigkeit dieser Einflüsse ist dies mit einer Diffusfeldentzerrung möglich.

Die HRTF ist durch die Messung einer BRIR mit einem Kunstkopf immer auch in dieser Messung enthalten. Nur dadurch ist eine Rekonstruktion des Schallfeldes bei der späteren Wiedergabe mit allen richtungsabhängigen Informationen möglich. In Gleichung 2.5, der Darstellung eines Systemteils einer BRIR, ist die HRTF in H_{Head} enthalten.

Neben dem Kunstkopf selbst hat auch der verwendete Lautsprecher Einfluss auf die Messung. Wie durch Slavik und Weinzierl (2008, S. 674) dargestellt, kann eine Freifeldentzerrung für die Hauptabstrahlrichtung des Lautsprechers hier nicht zur Kompensation dienen, da der Einfluss der individuellen Richtcharakteristik des Lautsprechers auf die Impulsantwort bestehen bleibt. In der Praxis kann also nur eine Diffusfeldentzerrung eine ansatzweise Näherung an ein Ideal bringen. In aller Regel werden aus diesem Grund weitgehend omnidirektionale Dodekaeder-Lautsprecher zur Gewinnung der Raumimpulsantworten verwendet.

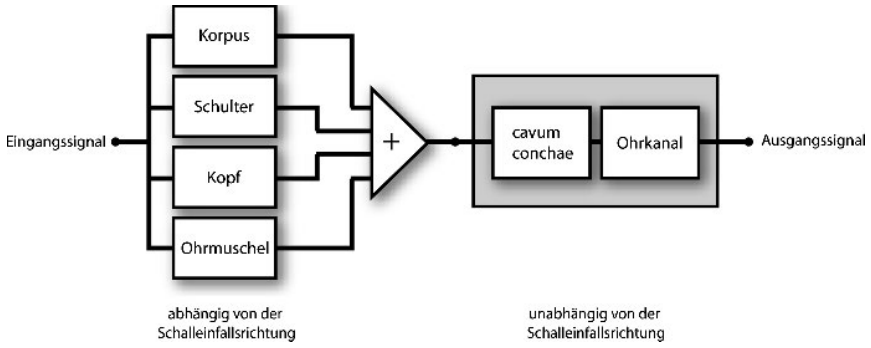


Abbildung 2.1.: Übertragungspfad des Außenohrs (Weinzierl, 2008, S. 586).

2.2.3. Binauralsynthese

Werden auf die zuvor beschriebene Weise binaurale Impulsantworten in einem Raum gemessen, liegen Daten vor, die geeignet sind, das Schallfeld, dem ein Hörer an der Messposition ausgesetzt war, zu rekonstruieren. Hierfür wird in der Binauralsynthese ein monaurales nachhallfreies Signal mit der BRIR getrennt für beide Ohren gefaltet und wiedergegeben. Als Schallwandler werden meistens Kopfhörer verwendet, jedoch ist auch eine Wiedergabe über Lautsprecher möglich. Diese Möglichkeit bringt das Problem des Übersprechens zwischen der linken und rechten Wiedergabeseite mit sich. Der Schallanteil, der für ein Ohr bestimmt ist, kann auch das andere erreichen (Møller, 1992, S. 172). Dieser Effekt kann mit Verfahren zur Unterdrückung des Übersprechens kompensiert werden (engl. Crosstalk Cancellation). Die Wiedergabe über Kopfhörer vermeidet dieses Problem, da die Schallwandlung direkt am Ohr

stattfindet und auf diese Weise eine klare Trennung möglich ist. Außerdem ist die Synthese des Schallfeldes für mehrere Hörer aus denselben Gründen nicht mit einer Lautsprecheranordnung zur binauralen Wiedergabe durchführbar. Deshalb wird hier nur auf die später auch verwendete Methode der Binauralsynthese mit Kopfhörern eingegangen.

Wenn die BRIRs, wie oben beschrieben, am geblockten Ohrkanal in der Ohrmuschel gemessen werden und das Aufnahmesystem für diesen Punkt entzerrt wird, muss die Wiedergabe auch unbeeinflusst von den Eigenschaften des Übertragungspfads vom Kopfhörer zum Ohrkanal stattfinden. Um dieses Ziel zu erreichen, muss auch dieser Übertragungsweg entzerrt werden. Dies geschieht durch eine Messung der Kopfhörer-Übertragungsfunktion (HpTF[6]) bei aufgesetztem Kopfhörer mit Miniaturmikrofonen im Ohrkanal. Die gewonnene Übertragungsfunktion wird nach spektraler Inversion und Nachbearbeitung als Kompensationsfilter verwendet (Slavik und Weinzierl, 2008, S. 674). Neben dieser individuellen Entzerrung des Kopfhörerfrequenzgangs ist jedoch auch eine allgemeine Entzerrung mithilfe eines Kunstkopfes möglich. Anstelle einer Messung der HpTFs für jeden Hörer einzeln, kann eine für den spezifischen Kopfhörer allgemeingültige Messung durchgeführt werden. Lindau und Brinkmann (2012) haben festgestellt, dass nicht-individuelle HpTFs bei der Wiedergabe zu einer realistischeren Wahrnehmung führen und somit keine individuelle Entzerrung für jeden Hörer durchgeführt werden muss.

Um eine korrekte HpTF zu ermitteln, müssen auch die Mikrofone des Kunstkopfes für die Messung entzerrt werden. Wie Møller (1992, S. 192) allerdings darstellt, ist eine Kompensation des Mikrofonfrequenzgangs nicht notwendig, solange bei der Messung der BRIR und der Messung der Kopfhörer-Übertragungsfunktion dieselben Mikrofone verwendet werden.

Eine einzelne gemessene BRIR ist immer auf eine Position im Raum festgelegt. Zum natürlichen Hörvorgang gehören jedoch auch Kopfbewegungen. Die Richtungsinformation aus der ein Hörer ein Schallereignis wahrnimmt, wird, wie zuvor beschrieben, aus dem Laufzeitunterschied des Schalls zwischen beiden Ohren gewonnen. Bewegungen des Kopfes ermöglichen Peilbewegungen, welche die Ortung einer Schallquelle zusätzlich verbessern, da der Abstand und somit auch die Laufzeit des Schalls zu beiden Ohren variiert wird (Blauert und Braasch, 2008, S. 88). Außerdem würde sich, bei Verwendung einer einzelnen BRIR für eine Kopfausrichtung, eine Schallquelle bei der Binauralsynthese ebenfalls bewegen, sobald der Hörer seinen Kopf dreht. Um diesem Problem entgegen zu kommen, werden an einer Position im Raum BRIRs für mehrere Kopfpositionen aufgezeichnet. Bei einer dynamischen Binauralsynthese kann nun, durch Bestimmung der Kopfposition durch einen Tracker, die für die jeweilige Kopfausrichtung richtige BRIR ausgewählt und für die Faltung mit dem nachhallfreien Signal verwendet werden. Abbildung 2.2 verdeutlicht diesen Ablauf. Als Freiheitsgrade sind horizontale und vertikale Kopfbewegungen bei fester

[6]Headphone Transfer Function

Position des Torsos (HATO[7]) möglich. Gemessen werden diese mit automatisierten Kunstkopfsystemen (HATS[8]), wie dem schon zuvor beschriebenen und an der TU Berlin entwickelten FABIAN (Lindau und Weinzierl, 2007), die in einem festen Messablauf BRIRs für alle benötigten Freiheitsgrade messen.

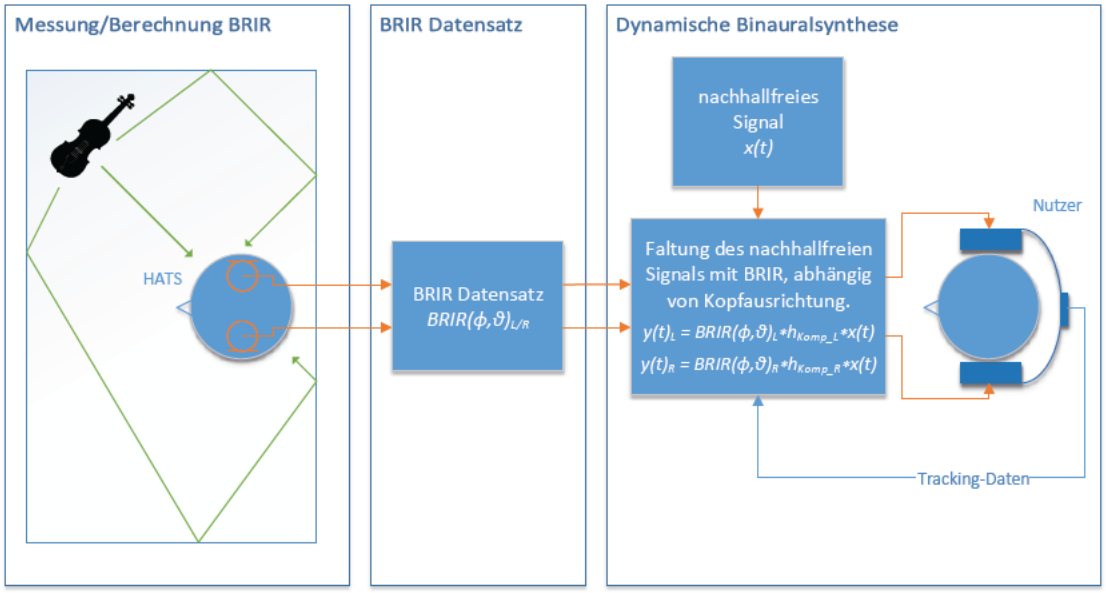


Abbildung 2.2.: Datenbasierte dynamische Binauralsynthese mit zeitlich unabhängiger Akquise und Reproduktion der Daten.

2.3. Akustische Simulation

Raumakustische Parameter und Impulsantworten für die Auralisation können bei bestehenden Räumen durch Messungen und bei der Planung von Räumen durch eine physikalische Simulation oder eine mathematische Computersimulation mit Modellen gewonnen werden (Ahnert und Tennhardt, 2008, S. 242). Aufgrund des hohen Aufwandes ein physikalisches Modell zu bauen und der größeren Flexibilität bei Änderungen am Modell, setzt sich die computergestützte Simulation immer weiter durch. Dabei sind zwei grundlegend verschiedene Berechnungsansätze zu betrachten. Zum Einsatz kommen einerseits Systeme, die ihre Berechnungen auf Grundlage der Wellentheorie durchführen und andererseits Techniken, die auf die Vorteile der geometrischen Akustik setzen.

Wellenbasierte Verfahren, wie die Finite Elemente Methode (FEM), teilen Räume in eine endliche Anzahl an Teilräumen auf, um für diese Lösungen der Wellengleichung zu berechnen. Diese Art der Simulation bietet laut Schröder (2011, S. 48)

[7]Head Above Torso Orientation
[8]Head and Torso Simulator

höchstmögliche Genauigkeit, da sie auch die Eigenheiten der Wellenausbreitung, wie die Schallbeugung und die Bildung von Moden, betrachtet. Sie erfordert jedoch enorme Rechenleistung und damit Rechenzeit, wenn das gesamte Frequenzband des Hörbereichs analysiert wird und die Größe der Räume ansteigt. Die Methode eignet sich gemäß Schröder in der Raumakustik also vor allem für kleine Räume und tiefe Frequenzen, deren Simulation mit anderen Methoden nicht möglich ist.

Für Frequenzen oberhalb der Schröderfrequenz $f = 2000 \cdot \sqrt{\frac{T}{V}}$ überlappen sich die modalen Strukturen eines Raums statistisch. Aus diesem Grund kann für sie die Methode der geometrischen Akustik angewendet werden, welche die modalen Strukturen des Schallfeldes vernachlässigt. Laut Lerch et al. (2009, S. 802-805) geht sie davon aus, dass sich Wellenamplituden nur langsam verändern und eine Wellenausbreitung mit ebenen Wellen stattfindet. Außerdem wird vorausgesetzt, dass die Wellenlänge der betrachteten Wellen klein ist, im Vergleich zu den geometrischen Abmessungen der im Raum vorkommenden Flächen. Unter diesen Vorraussetzungen können Schallwellen durch Strahlen, die von der Quelle ausgesendet werden, simuliert werden. Jeder Strahl fungiert dabei als Äquivalent für eine ebene Wellenfront und ihre Ausbreitungsrichtung.

Für die Berechnung von Lösungen der geometrischen Akustik kommen verschiedene Methoden zum Einsatz. Hier sollen die Methoden der Spiegelschallquellenberechnung und des Ray Tracings näher erläutert werden. Zusammen angewendet bilden sie die Grundlage einer hybriden Berechnungsmethode, welche in aktuellen Softwareimplementierungen zum Einsatz kommt. Dieses Verfahren wurde für die Entwicklung der Versuchsumgebung mit der eingesetzten Software RAVEN ebenfalls verwendet.

2.3.1. Deterministische Spiegelschallquellenmethode

Die Spiegelschallquellenmethode ist ein Weg, das Schallfeld, beschrieben durch ein Strahlenmodell, für einen Empfänger im Raum zu berechnen. Hierbei handelt es sich zunächst um eine einfach auszuführende geometrische Berechnung. Für diese Berechnung werden Spiegelungen der Quelle an den Begrenzungsflächen des Raums konstruiert. Die Länge des Weges von der Quelle zum Empfänger über die Reflexion an der Fläche ist in der direkten Entfernung von der Spiegelquelle zum Empfänger enthalten. Abbildung 2.3 verdeutlicht das Modell. Diese Art der geometrischen Konstruktion bedingt, dass nur Schallstrahlen betrachtet werden, die den Empfänger auch erreichen. Reflexionen, die den Empfänger nie erreichen, werden nicht berechnet (Chéenne, 2008, S. 220). Spiegelschallquellen, die durch die einfache Spiegelung und somit einfache Reflexion an einer Fläche zustande kommen, werden Spiegelschallquellen erster Ordnung genannt. Weitere Spiegelungen an den übrigen Begrenzungsflächen entsprechen dann Spiegelschallquellen höherer Ordnung. Die Anzahl der berechneten Spiegelschallquellen steigt also mit der Anzahl der Raumbegrenzungsflächen N und der gewünschten Ordnung i der Spiegelschallquellen.

Die Berechnungszeit ist proportional zu N^i (Ahnert et al., 2008, S. 1371). Hinzu kommt, dass für jede Spiegelschallquelle noch bestimmt werden muss, ob sie für den Empfänger sichtbar, also relevant ist oder von anderen Raumbegrenzungen verdeckt wird. Aus diesen Eigenschaften ergibt sich auch der Nachteil der Methode. Sie ist für rechteckige Räume sehr akkurat, wird aber für komplexere Räume mit vielen Begrenzungsflächen sehr rechenintensiv. Deshalb wird sie in heutiger Simulationssoftware meistens nur für sehr einfache rechteckige Räume oder für die Berechnung von Reflexionen geringer Ordnung verwendet (Rindel, 1995, S. 2). Letzteres kommt laut Rindel (1995) beim Design von Lautsprecher-Systemen oder in Kombination mit Ray-Tracing-Verfahren für die Berechnung von Reflexionen höherer Ordnung in Frage.

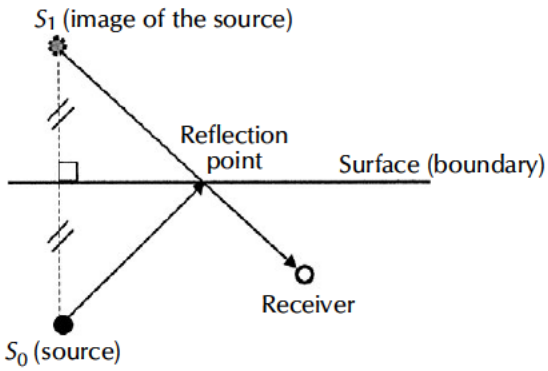


Abbildung 2.3.: Spiegelschallquellenmodell (Chéenne, 2008, S. 221).

Die Konstruktion der Impulsantwort bei Anwendung der Spiegelschallquellenmethode geschieht analog zu Gleichung 2.5. Die Impulsantwort eines einzelnen Übertragungspfades wird also aus der Quellencharakteristik, der Luftabsorption, den Reflexionsfaktoren der Oberflächen und der Empfängercharakteristik gewonnen. Die Raumimpulsantwort wird durch Summierung der einzelnen Antworten der Übertragungspfade im Zeitbereich berechnet. Für die Berechnung einer binauralen Impulsantwort wird für die Empfängercharakteristik ein HRTF-Datensatz eingesetzt. Abhängig vom Einfallswinkel des Schalls eines einzelnen Reflexionspfades wird die passende HRTF mit der Übertragungsfunktion des Schallstrahls multipliziert. Dadurch werden zwei neue Übertragungsfunktionen, jeweils für das linke und rechte Ohr und somit nach der Transformation in den Zeitbereich die BRIR gewonnen.

Die Spiegelschallquellenmethode beinhaltet nur die Reflexion und Absorption von Schallstrahlen an den Raumbegrenzungsflächen. Die Streuung wird nicht berechnet, da die Berechnung nicht mit der Methode umzusetzen ist. Allerdings können moderne Computersysteme die Beugung des Schalls mit berücksichtigen. Hierfür werden, wie von Schröder (2011, S. 56) beschrieben, dem Modell für jede Kante, die einen Strahl blockiert, weitere Spiegelschallquellen hinzugefügt, die in die

Berechnung mit einfließen. Ein Algorithmus bestimmt, ob die dadurch entstehenden Quellen für den Empfänger sichtbar sind und ob sie berechnet werden müssen.

2.3.2. Stochastisches Ray Tracing

Basierend auf dem aus der Computer-Grafik bekannten Verfahren (Goldstein und Nagel, 1971), nimmt das Ray-Tracing-Verfahren für die geometrische Akustik die Ausbreitung von Schallwellen ebenfalls in Form von Strahlen an. Von einer Quelle werden diese Strahlen in zufälliger Richtung ausgesandt. Jeder einzelne Strahl wird verfolgt, bis er auf den Empfänger trifft oder durch die Absorption der Luft und der Raumbegrenzungsflächen ein festgelegtes Energieniveau unterschritten hat.

Auch das Ray-Tracing-Verfahren schließt wieder die in Gleichung 2.5 aufgeführten Einflüsse des als LTI-System betrachteten Raums mit ein: Die Energie der ausgesendeten Strahlen ist von der Richtcharakteristik der Quelle abhängig. So kann neben einer ideal kugelförmigen Quelle auch die spezifische Charakteristik eines Lautsprechers oder Instruments verwendet werden. Die Richtcharakteristik enthält in ihren Auflösungsgrenzen Daten für jeden Raumwinkel und Frequenzbereich. Gröbere Auflösungen können vom Computer interpoliert werden. Die Energie der ausgesandten Strahlen wird durch die Richtcharakteristik, abhängig von der Frequenz, beeinflusst.

Auf dem Übertragungsweg wirken Luftabsorption und Absorption durch Hindernisse im Ausbreitungsweg auf die Energie und die zeitliche Verzögerung. Die Frequenzabhängigkeit von Reflexion und Absorption einer Schallwelle in einem Medium führt im Ray-Tracing-Verfahren dazu, dass für jedes Frequenzband eine eigene Berechnung durchgeführt werden muss. Somit hat die Anzahl der Frequenzbänder (Terz- oder Oktaveinteilung), die Anzahl der ausgesandten Strahlen und die Größe des Raums mit seinen Oberflächeneigenschaften Einfluss auf die benötigte Rechenzeit in der Simulation.

Trifft ein Strahl auf eine Oberfläche, ist die Entscheidung, ob er reflektiert, gestreut oder absorbiert wird, abhängig von der Frequenz und der Oberflächenbeschaffenheit. Ist die Wellenlänge groß gegenüber der Struktur der Oberfläche, wird der Strahl reflektiert und folgt dem aus der Optik bekannten Gesetz „Einfallswinkel gleich Ausfallswinkel“. Ist seine Wellenlänge gegenüber der Struktur klein, wird der Strahl gestreut (Chéenne, 2008, S. 220). Dies geschieht, da sich die einzelnen unregelmäßigen Teile der Feinstruktur für kleine Wellenlängen wiederum wie glatte, reflektierende Oberflächen verhalten (Kuttruff, 2009, S. 59). In Simulationsprogrammen wird nicht für jede Oberfläche eine Feinstruktur modelliert, sondern ein Koeffizient für den Grad der Streuung und einer für die Absorption in Oktav- oder Terzbändern für ein Oberflächenmaterial festgehalten (Schröder et al., 2007). Ob ein Strahl gestreut wird, entscheidet die Software über einen Zufallsprozess. Je nach Grad der Streuung und der Absorption verringert sich die Energie des Strahls.

Im Gegensatz zum Spiegelschallquellenverfahren besteht der Empfänger nicht aus einem geometrischen Punkt im Raum, sondern aus einem meist kugelförmigen Detektor mit einem definierten Radius. Für einen einzelnen Punkt im Raum würde die Wahrscheinlichkeit eines Zusammentreffens eines Strahls mit dem Empfänger gegen Null gehen (Schröder, 2011, S. 61). Der Radius des Detektors ist entscheidend für die Genauigkeit der Berechnung, da ein größerer Radius mehr Schnittpunkte mit Strahlen, aber auch ein anderes Energieniveau des Schallfeldes erreicht.

Konstruktion der Impulsantwort in RAVEN

Die Konstruktion der Impulsantwort beim Ray-Tracing-Verfahren soll an dieser Stelle anhand des durch Schröder (2011) ausführlich dokumentierten Verfahrens der Software RAVEN erläutert werden, da diese in der entwickelten Versuchsumgebung zum Einsatz kommt. Auf weitere Details der Simulations-Software und ihrer Anwendung wird in Abschnitt 3.1 eingegangen.

Alle Strahlen, die einen Schnittpunkt mit dem Detektor aufweisen, werden in einem dreidimensionalen Histogram festgehalten. In diesem ist der durch die Wegstrecke durch den Raum verzögerte Zeitpunkt des Eintreffens eines jeden Strahls auf der Abszisse und das Frequenzband auf der Ordinate notiert. Auf der Applikate wird die Energie festgehalten. Dieses Histogramm enthält alle Informationen, um eine RIR zu konstruieren. Es kann als zeitliche Energieverteilung für ein einzelnes Frequenzband oder als spektrale Kurz-Zeit-Energiedichte für einen Zeitpunkt interpretiert werden.

RAVEN verwendet ein stochastisches Ray-Tracing-Verfahren und bedient sich hierbei Poisson-verteilter Zufallssequenzen, die mit der Energieverteilung gewichtet werden. Parameter, wie die Nachhallzeit, können direkt aus den Werten des Histogramms berechnet werden. Eine hieraus gewonnene Impulsantwort würde mit der geringen zeitlichen Auflösung jedoch laut Schröder nicht für eine Auralisation ausreichen. Um eine ausreichend hohe Auflösung zu erlangen, müssten sehr viele Strahlen ausgesendet werden, da sich ja nur ein Bruchteil der Stahlen mit dem Detektor im Raum schneidet. Die Berechnungszeit wäre dadurch nicht mehr vertretbar. Aus diesem Grund wird die Impulsantwort aus den Daten und den Zufallssequenzen synthetisiert.

Zur Gewinnung einer **monauralen RIR** wird gemäß Schröder (2011, S. 71) in RAVEN eine Poisson-verteilte Zufallssequenz von positiven und negativen Dirac-Impulsen generiert. Diese wird in den Frequenzbereich transformiert und mit Hoch- und Tiefpassfilter-Kombinationen in Oktav- oder Terzbänder aufgeteilt. Die entstehenden zehn bzw. 31 Spektren werden zurück in den Zeitbereich transformiert und mit der Hüllkurve der Energieverteilung aus dem Energiehistogramm gewichtet. Die resultierenden einzelnen synthetisierten Impulsantworten werden zu der gesuchten Raumimpulsantwort aufsummiert.

Bei der Gewinnung einer **binauralen RIR** muss die Charakteristik des Empfängers und somit auch die Schalleinfallsrichtung für jeden Kanal getrennt beachtet werden. Als Charakteristik für den Empfänger wird ein HRTF-Datensatz verwendet. Zu diesem Zweck teilt RAVEN den sphärischen Detektor laut Schröder (2011, S. 72) mit einem kopfbezogenen Koordinatensystem in diskrete Richtungsgruppen (RG) auf. Für jeden Zeitabschnitt werden die Schnittpunkte der eintreffenden Schallstrahlen mit dem Detektor bestimmt und in dem bereits beschriebenen Energiehistogramm festgehalten. Pro Richtungsgruppe und Zeitabschnitt wird nun die Energie der einzelnen eintreffenden Strahlen über alle Frequenzbänder aufsummiert. Die Summen für jede Gruppe werden auf die Gesamtenergie des Zeitabschnitts normiert. Die resultierende Wahrscheinlichkeit für die Schalleinfallsrichtung zwischen 0 und 1 wird kumuliert über die einzelnen Richtungsgruppen je Zeitabschnitt in einem Histogramm festgehalten.

Wie bei der Gewinnung der monauralen RIR wird auch hier eine Poisson-verteilte Dirac-Folge generiert. Zusätzlich zu dieser wird jetzt laut Schröder noch für jeden Dirac-Stoß der Folge eine gleichverteilte Zufallszahl zwischen {0,1} erzeugt. Durch den Vergleich der Zufallszahl mit der kumulierten Wahrscheinlichkeit wird die Richtungsgruppe pro Zeitabschnitt für die aktuelle Einfallsrichtung des Schalls ausgewählt. Höhere Wahrscheinlichkeiten werden dabei öfter gewählt als kleinere. Wird beispielsweise Richtungsgruppe 1 mit 25% Wahrscheinlichkeit getroffen und RG 2 mit 75% wird die kumulierte Wahrscheinlichkeit mit den Werten 0 ... 0,25 für RG 1 und 0,26 ... 1 für RG 2 festgehalten. Beim Vergleich mit einer Zufallszahl ist die Wahrscheinlichkeit höher, dass diese in den Bereich der Richtungsgruppe 2 fällt. Durch den Vergleich mit einer Zufallszahl, anstelle einer direkten Wahl der Richtungsgruppe anhand der höchsten Wahrscheinlichkeit, wird zwar die Richtung für jeden Zeitbereich proportional zur Wahrscheinlichkeit bestimmt, es ergibt sich jedoch auch eine Streuung auf geringere Wahrscheinlichkeiten, die laut Schröder (2011, S. 74) zu einem homogeneren Diffusschall führt.

Die zu den gewählten Gruppen aufgrund ihrer Raumrichtung passenden HRIRs[9] werden mit dem Dirac-Stoß aus der Poisson-verteilten Folge des gerade betrachteten Zeitabschnitts gefaltet, was aufgrund der idealen Dirac-Folge einer reinen Zeitverschiebung der HRIRs entspricht. Die Summe dieser zeitversetzten HRIRs ergibt zwei Noise-Sequenzen, jeweils für den rechten und linken Kanal, welche die räumliche Information für einfallende Schallstrahlen enthalten. Diese Sequenzen werden nach Schröder zuletzt mit einem von-Hann-Fenster in gleich große Zeitabschnitte aufgeteilt und in den Frequenzbereich transformiert. Die Kurzzeitspektren aller Richtungsgruppen werden je Zeitabschnitt getrennt aufsummiert. Jeder Abschnitt der aufgeteilten Noise Sequenz wird daraufhin mit dem zum Zeitpunkt passenden Spektrum gewichtet. Nach der Rücktransformation in den Zeitbereich werden die Fragmente zur finalen BRIR zusammengesetzt.

[9] Head Related Impuse Responses

2.3.3. Hybride Methode

Die in Abschnitt 2.3.1 beschriebene Spiegelschallquellenmethode kann aufgrund ihrer exakten Zeitauflösung sehr akkurate Ergebnisse für die Lokalisation von Schallquellen im virtuellen Schallfeld erzielen (Schröder et al., 2007, S. 1). Für große Räume und viele Oberflächen steigt die benötigte Rechenzeit jedoch stark an. Im Gegensatz dazu ist mit dem Ray Tracing-Verfahren auch die Simulation eines Schallfeldes in solchen Räumen und unter Einbezug der Streuung der Strahlen möglich. Hier muss jedoch die Impulsantwort für die Auralisation aufgrund der geringen Auflösung des resultierenden Reflektogramms mit stochastischen Prozessen synthetisiert werden. Die hybride Methode vereint die Vorteile der Spiegelschallquellenmethode und des Ray Tracings. Die akkurate Berechnung der deterministischen Spiegelschallquellenmethode wird für die frühen Reflexionen geringer Ordnung verwendet. Dadurch bleibt die, für die Lokalisationsfähigkeit des menschlichen Gehörs benötigte, exakte zeitliche Auflösung der ersten Reflexionen erhalten (Schröder, 2011, S.48). Das stochastische Ray Tracing wird für den diffusen Anteil des Schallfeldes, der die Reflexionen höherer Ordnung und die Streuung mit einbezieht, verwendet. Im diffusen Schallfeld treffen die Schallwellen definitionsgemäß aus zufälliger Richtung auf den Empfänger und sind für den Raumeindruck verantwortlich. Da die exakte Richtungsinformation also nicht entscheidend ist, kann das Diffusfeld mithilfe von Zufallsprozessen im Ray Tracing-Algorithmus berechnet werden.

Die Zusammenführung der Ergebnisse beider Berechnungsmethoden setzt voraus, dass das Energieniveau der Ergebnisse beider Methoden aufeinander abgestimmt wird und kein Reflexionspfad doppelt vorkommt. Mit dem Spiegelschallquellenmodell werden Strahlen mit reinen Spiegelungen bis zu einer festgelegten Ordnung berechnet. Durch das Ray Tracing werden diffus reflektierte Strahlen und die rein gespiegelten Strahlen erfasst, welche eine höhere Ordnung als das Spiegelschallquellenmodell aufweisen. Um das Energieniveau beider Modelle aneinander anzupassen, wird laut Schröder (2011, S. 75) mit jeder Methode der Direktschall im Freifeld für die Sender-Empfänger-Distanz berechnet und für die Kalibrierung der Impulsantworten verglichen. Für die finale RIR bzw. BRIR werden die Antworten beider Systeme zu einer gemeinsamen, wie in Abbildung 2.4 dargestellten, Impulsantwort addiert.

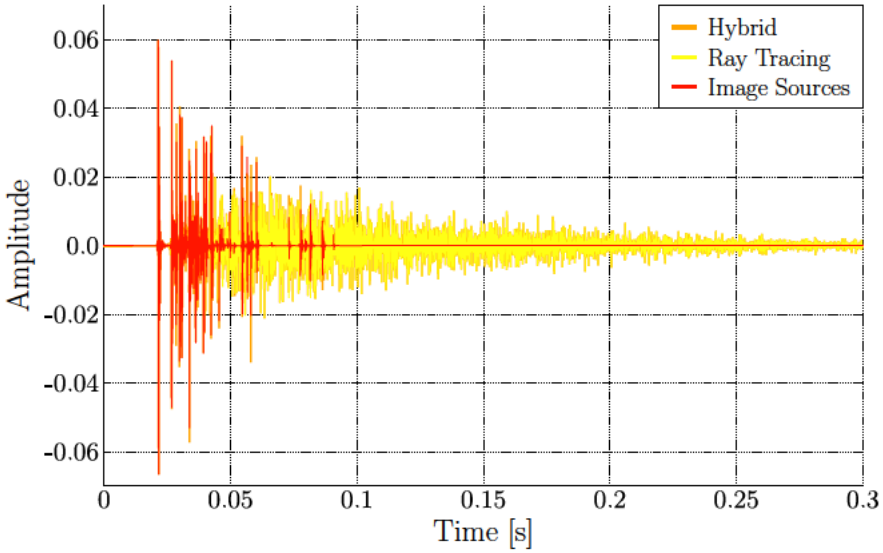


Abbildung 2.4.: Summe der Impulsantworten gewonnen mit stochastischem Ray Tracing und der Spiegelschallquellenmethode in RAVEN. (Schröder, 2011, S. 76)

3. Methode

Auf Basis der in Kapitel 2 beschriebenen Vorraussetzungen und Möglichkeiten wurde eine Versuchsumgebung für die Echtzeit-Auralisation einer VAE für vier Musiker im RAR entwickelt. Unter Berücksichtigung der Vorgaben für die später vorgesehenen Versuche schließt diese die Erzeugung aller nötigen Daten und den technischen Aufbau im Labor ein. Der Versuchsumgebung wurde der Titel *Virtual Stage Environment* (VSE) gegeben, welcher im Folgenden als Kennzeichnung für das Gesamtsystem verwendet wird.

Folgende Vorgaben und Ziele wurden für die Entwicklung der Versuchsumgebung definiert:

- Offline BRIR Berechnung für VAE mit RAVEN
  - Umsetzung von 14 Raummodellen in SketchUp
  - Implementierung der benötigten Richtcharakteristiken
  - Automation der BRIR Berechnung
  - Feststellen geeigneter Berechnungsparameter
- Technischer Aufbau der Versuchsumgebung
  - Geeigneter Aufbau der Aufnahme-/Wiedergabekette
  - Gleichzeitiges Laden mehrerer VAEs für schnellen Wechsel
  - Beachtung der Audio-Latenz
  - Kalibrierung des Direkt-/Diffusschallverhältnisses
  - Automation des Versuchsablaufs
- Evaluation der Versuchsumgebung

Die Entwicklung der gesamten Versuchsumgebung kann somit in zwei Teilbereiche gegliedert werden. Den ersten Komplex bildet die Berechnung der BRIRs, welche die Grundlage für die virtuellen akustischen Umgebungen bilden. Diese BRIRs werden ausschließlich für den Diffusschall und die frühen Reflexionen berechnet. Der Direktschall wird aus der Simulation ausgeschlossen, da die Versuche im reflexionsarmen Raum mit offenen Kopfhörern stattfinden sollen. Der Direktschall erreicht die Musiker also direkt von ihren Instrumenten. Dem Schallfeld am Ohr des Empfängers soll nur die Antwort des virtuellen Raums auf diesen Direktschall durch die Auralisation hinzugefügt werden.

Für einen Versuchsaufbau mit vier Musikern müssen BRIRs für 16 Sender- und Empfängerkombinationen erzeugt werden, um das von jedem Instrument individuell erzeugte Schallfeld an den vier Empfängerpositionen rekonstruieren zu können. Die wichtigsten Merkmale bei der Berechnung sind die exakte Positionierung und Ausrichtung der Quellen und Empfänger zueinander, die Implementierung ihrer Richtcharakteristiken und der virtuelle Raum. Hinzu kommen Parameter für die Berechnung, welche zum einen die Genauigkeit und zum anderen die benötigte Rechenzeit beeinflussen. Hier war ein Kompromiss zu finden, der eine korrekte und plausible Auralisation ermöglichte und gleichzeitig in einem geeigneten Zeitraum durchführbar war.

Der zweite Teil besteht im Aufbau des Laborversuchs. Die zuvor berechneten BRIRs sollen computergestützt in der Auralisation des Schallfeldes des virtuellen Raums, kombiniert mit dem Direktschall in der Realität, für die vier gemeinsam musizierenden Musiker verarbeitet werden. Hier wurde ein geeigneter Aufbau der Aufnahme- und Wiedergabekette für die in Echtzeit zu verarbeitenden Signale entwickelt. Die Mikrofonierung der Instrumente, die Signaldistribution, die Faltung mit den BRIRs und die Wiedergabe über geeignete Kopfhörer inklusive der Aufnahme der Kopfausrichtung bilden die Stationen der Signalverarbeitungskette. Als wichtiger Faktor wurde außerdem die Kalibrierung der Systems betrachtet, die erheblichen Einfluss auf die Authentizität der Auralisation hat. Für die computerinterne Verarbeitung der Signale wurde eine Softwareumgebung entwickelt, die unter Berücksichtigung der Signallatenz und -qualität, die Möglichkeit bietet, online zwischen verschiedenen Räumen zu wechseln und die einzelnen Schritte der Versuchsdurchführung zu automatisieren.

Zum Abschluss der Entwicklung des Versuchsaufbaus wurde ein Vorversuch durchgeführt, welcher die Evaluation der Funktionalität der Umgebung und der entwickelten Verfahren sowie der Plausibilität der dargebotenen VAEs zum Ziel hatte. Hierfür wurde ein Interviewleitfaden erstellt, anhand dessen die Probanden des Versuchs befragt wurden.

3.1. Konzeption der virtuellen akustischen Umgebungen und Berechnung der zugehörigen BRIRs

Der erste Teil der Realisierung der virtuellen akustischen Umgebungen besteht in der Simulation der Schallfelder der virtuellen Räume in einer Simulationssoftware. Auf diese Weise werden die für die darauf folgende Auralisation benötigten binauralen Impulsantworten erzeugt. Die Berechnung der BRIRs für die Auralisation der VAEs wurde mit der Software RAVEN (Aspöck et al., 2014; Schröder und Vorländer, 2011) umgesetzt. RAVEN wurde am Institut für technische Akustik der RWTH Aachen

entwickelt. Es handelt sich um eine hybride Raumakustik-Simulationssoftware, die, wie zuvor im Grundlagenkapitel beschrieben, die deterministische Spiegelschallquellenmethode und das stochastisches Ray Tracing für die Berechnung von raumakustischen Parametern und die Auralisation von virtuellen Räumen einsetzt. Diese kann mit RAVEN in Grenzen in Echtzeit erfolgen. Außerdem ist eine offline-Berechnung von BRIRs möglich. Diese BRIRs können später für eine Auralisation in Echtzeit in einer unabhängigen Software mit nachhallfreien Signalen gefaltet werden.

Ein großer Vorteil von RAVEN ist die direkte Einbindung in die Software SketchUp durch ein Plugin, welches die Berechnung von raumakustischen Parametern und die Auralisation direkt aus dem in Sketchup gezeichneten CAD-Modell ermöglicht. Die Plugins greifen auf das eigentliche RAVEN-Framework zu, welches außerdem durch eine grafische Benutzeroberfläche (GUI[1]) zu bedienen ist. Die zusätzliche Möglichkeit die Konfiguration von RAVEN mit Matlab vorzunehmen und aus Matlab heraus die Berechnung zu starten wurde für die Entwicklung der Versuchsumgebung verwendet, um eine große Anzahl von unterschiedlichen Berechnungen weitestgehend dynamisch und automatisiert durchzuführen. Hierfür wurde ein Skript mit zugehörigen Funktionen in Matlab entwickelt, welches in einer eigenständig ablaufenden Routine alle vorgegebenen Berechnungen durchführt.

Für die Simulation und Berechnung der BRIRs wurden 14 Raummodelle erstellt und für RAVEN optimiert. In diesen Modellen wurden für die Musiker virtuelle Quellen und Empfänger platziert und korrekte Richtcharakteristiken implementiert. Für die möglichen Kopfbewegungen der Musiker wurden Freiheitsgrade definiert. Die daraufhin benötigten Berechnungen der BRIRs für die entsprechenden Ausrichtungen der Empfänger im virtuellen Raum wurden in die Berechnungsroutine in Matlab integriert. Mehrere Funktionen erlauben hier den Export in diverse Dateiformate und -container und ermöglichen den Einsatz der BRIRs in verschiedenen Anwendungsgebieten. Darunter ist auch das SOFA[2]-Dateiformat, welches vom später verwendeten Sound Scape Renderer (SSR) für das Rendering der VAE mit den berechneten BRIRs verwendet wurde. Bei der Berechnung und dem Export der BRIRs kamen verschiedene Softwareprodukte (RAVEN, SketchUp, SSR) mit unterschiedlichen Koordinatensystemen zum Einsatz. Dies galt es bei der Weiterverarbeitung und Verwendung der Daten zu beachten. Vor der endgültigen Berechnung der BRIRs mussten alle für die Simulation mit RAVEN benötigten Parameter gewählt oder berechnet werden. Die Berechnungen wurden in das Matlab-Skript integriert.

[1] Graphical User Interface
[2] Spatially Oriented Format for Acoustics: http://www.sofaconventions.org/

3.1.1. Simulation mit RAVEN

Für die Raumakustik Simulation in RAVEN wird zunächst das Modell des zu berechnenden Raums in SketchUp gezeichnet. Das RAVEN Plugin in SketchUp bietet hier die Möglichkeit die Oberflächen des Modells direkt in SketchUp mit Materialen zu belegen, welche akustische Eigenschaften, wie Absorptions- und Streugrade in Oktav- oder Terzbändern, beinhalten können. Nach Fertigstellung des Modells können Berechnungen von raumakustischen Parametern, wie z.B. der Nachhallzeit, des Stärkemaßes und des Sprachverständlichkeitsindexes, direkt aus SketchUp heraus getätigt werden. Nach erfolgreicher Kalkulation werden die Ergebnisse in Diagrammen und als Colormaps in SketchUp angezeigt. Ebenso bietet sich die Möglichkeit Auralisationen direkt aus SketchUp heraus zu berechnen und Kamerafahrten mit passender Auralisation zu programmieren.

Für die externe Durchführung der Simulationen durch Matlab oder die RAVEN GUI wird das erstellte Modell als AC3D-Datei[3] und die Parameter für die Berechnung sowie Quell- und Empfängerpositionen als RPF-Datei[4] aus SketchUP exportiert. Beide Dateien sind als Textdateien angelegt, wodurch sie nach dem Export im Texteditor bearbeitet werden können. In der RAVEN-Installation ist ein Matlab-Framework enthalten, welches unter anderem Methoden und Funktionen zur Bearbeitung von Modell- und Berechnungsparametern, Quell- und Empfängerpositionen bietet. Grundsätzlich greifen die Funktionen immer auf die RPF-Datei des Projekts zu und manipulieren diese. Für die Simulation wird RAVEN aus Matlab heraus als Windows-Kommandozeilenprogramm mit der Projektdatei (RPF) als Parameter gestartet. Abschnitt 3.1.7 beschreibt das VSE-Matlab-Skript, welches für die Versuchsumgebung programmiert wurde, um die Berechnungen für alle Modelle automatisiert durchzuführen.

3.1.2. Koordinatensysteme

Für die Entwicklung der Versuchsumgebung werden drei Softwareprodukte eingesetzt, die unterschiedliche Koordinatensysteme verwenden. Bei der Programmierung und der Weiterverarbeitung der Daten muss dieser Umstand berücksichtigt werden. SketchUp und der Sound Scape Renderer (SSR) verwenden ein rechtshändiges Koordinatensystem, wie in Abbildung 3.1a dargestellt. Das Koordinatensystem in RAVEN dagegen ist in Abbildung 3.1b aufgezeichnet. Die Vorschrift für die Überführung von SketchUp- in RAVEN-Koordinaten lautet:

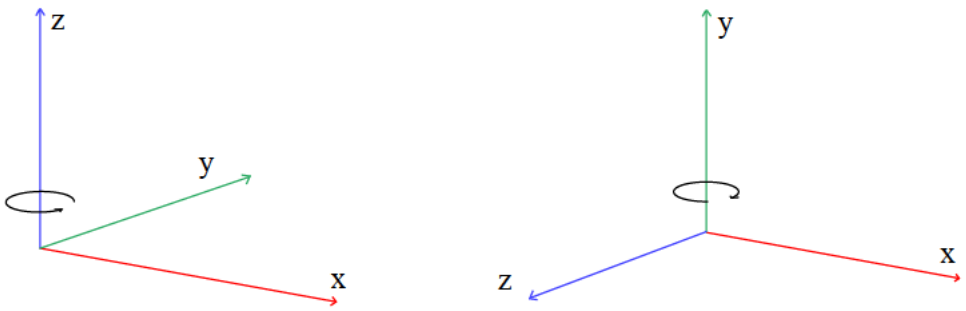
$$\begin{bmatrix} x \\ y \\ z \end{bmatrix}_{RAVEN} = \begin{bmatrix} x \\ z \\ -y \end{bmatrix}_{SketchUp} \tag{3.1}$$

[3]Dateiformat für dreidimensionale Computergrafik Modelle (*.ac)
[4]RAVEN-Project-Datei mit RAVEN-Projektparametern (*.rpf)

Umgekehrt kann das Koordinatensystem von RAVEN- in SketchUp-Koordinaten überführt werden:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix}_{SketchUP} = \begin{bmatrix} x \\ -z \\ y \end{bmatrix}_{RAVEN} \tag{3.2}$$

In der Abbildung dargestellt ist außerdem die Drehrichtung in der Horizontalebene um die vertikale Achse. Im Fall von RAVEN verläuft eine Drehung um die y-Achse in der x-z Ebene im Uhrzeigersinn positiv. Mit dem Matlab-Skript berechnete BRIRs werden bei positiven Winkeln für Drehwinkel im Azimuth im Uhrzeigersinn berechnet. Der SSR und das zugehörige SOFA-Dateiformat erwarten jedoch für positive Winkel eine Drehung gegen den Uhrzeigersinn. Das Skript und die Exportfunktionen beachten diese unterschiedlichen Konventionen und konvertieren die Winkel und Koordinaten entsprechend.



(a) SketchUp/SSR (b) RAVEN

Abbildung 3.1.: Koordinatensysteme von SketchUp und RAVEN

3.1.3. Modelle

Die für diese Versuchsumgebung verwendeten 14 Raummodelle wurden bereits für die Versuche in Schärer Kalkandjiev und Weinzierl (2015) und (2013) verwendet. Die Räume sind in Tabelle 3.1 aufgeführt und in Abbildung 3.3 und 3.4 dargestellt. Für jeden Raum wurde eine Abkürzung vergeben, welche die Arbeit in den Softwareumgebungen vereinfacht. Im weiteren Verlauf dieser Arbeit werden diese Abkürzungen verwendet.

Die Räume stellen typische Aufführungsorte für westeuropäische Kunstmusik dar und sind von existierenden Räumen inspiriert. Die Auswahl der Modelle bildet eine große Bandbreite unterschiedlicher Parameter der Raumakustik ab. Durch Schärer Kalkandjiev und Weinzierl (2015) wurden dabei folgende Parameter untersucht: EDT, RT, C_{80}, G, BR, ST_{early}, ST_{late}, G_{e}, G_{l}, G_{125}. Die erreichte große Streuung

dieser raumakustischen Parameter für die ausgewählten Modelle wird begleitet durch die Variation im Volumen, welches zwischen 900 m^3 und 21661 m^3 liegt. Dies führt zusammen mit unterschiedlichen Oberflächenmaterialien zu Nachhallzeiten zwischen 0,94 und 3, 17 s. Die Streuung dieser Werte wird in Abbildung 3.2 verdeutlicht. Die Räume CHU und CON1 wurden jeweils mit unterschiedlichen Absorptionskoeffizienten und daraus resultierend unterschiedlicher Nachhallzeit in zwei Versionen (a/b) in den Versuch aufgenommen. So sind im Hörversuch auch Vergleiche zwischen den Räumen bei unveränderter Raumgeometrie, aber veränderten Oberflächeneigenschaften möglich. Zusätzlich zu den 11 Räumen aus Schärer Kalkandjiev und Weinzierl (2015) wurde der Raum CCC aus Schärer Kalkandjiev und Weinzierl (2013) simuliert. Er kann durch seinen Aufbau ohne Decke als Raumbegrenzung die Raumauswahl um einen charakteristischen Typ erweitern.

Die ursprünglich in Schärer Kalkandjiev und Weinzierl (2013, 2015) verwendeten Modelle wurden mit der Software EASE[5] konstruiert. RAVEN weist einige Unterschiede in der CAD-Konstruktion im Vergleich zu EASE auf. Oberflächen dürfen für RAVEN z.B. nur von einer Seite mit einem Material belegt werden. Außerdem kann der Import von EASE-Modellen in SketchUp zu Komplikationen führen. Schon kleine Unterschiede in den Koordinaten von Punkten und Linien führen zu Verzerrungen in Flächen und dadurch einer erhöhten Anzahl von Polygonen, wodurch die Rechenzeit verlängert wird. Eine zu große Anzahl an Flächen und in Relation zum betrachteten Frequenzbereich sehr kleine Flächen führen, wie in Abschnitt 2.3.1 und 2.3.2 beschrieben, ebenfalls zu längeren Rechenzeiten. Um die Kompatibilität und Simulationsgeschwindigkeit zu optimieren, wurden die Modelle für die Versuchsumgebung neu konstruiert. Durch die Neubearbeitung konnte die benötigte Rechenzeit deutlich reduziert werden.

[5]Electro-Acoustic Simulator for Engineers

Raum	Abkürzung	Volumen/m^3	Fläche/m^2	Bühne/m^2	T30/s
Auditorio Nacional, Madrid	ANC	5714,50	2976,24	109,60	1,64
Cloitre, Forcalquier	CCC	10046,57	3001,60	-	0,94
Gulbenkian Hall, Lissabon	GGA	9745,19	3561,37	95,16	1,23
Kammersaal 1	CHA1	2334,81	1470,92	56,00	0,78
Kammersaal 2	CHA2	3232,76	1879,87	85,28	1,77
Kirche a	CHUa	12529,96	4402,20	67,31	3,17
Kirche b	CHUb	12529,96	4402,20	67,31	2,28
Konzertsaal 1 a	CON1a	21661,05	5192,70	108,90	3,08
Konzertsaal 1 b	CON1b	21661,05	5192,70	108,90	1,91
Konzertsaal 2	CON2	10261,58	3704,58	199,53	1,76
Oper	OPR	14862,00	7139,63	96,80	1,13
Eroicasaal Palais Lobkowitz, Wien	PLE	900,17	603,30	30,06	1,64
Teatre Jean Vilar, Paris	TJV	11175,00	3923,00	150,24	0,69
Wigmore Hall, London	WMH	2772,74	1395,10	33,48	1,38

Tabelle 3.1.: Auflistung der verwendeten Räume mit mittlerer Nachhallzeit T30, Raumvolumen, Gesamtoberfläche der Raumbegrenzungen, Bühnenfläche und Abkürzung der Raumbezeichnung.

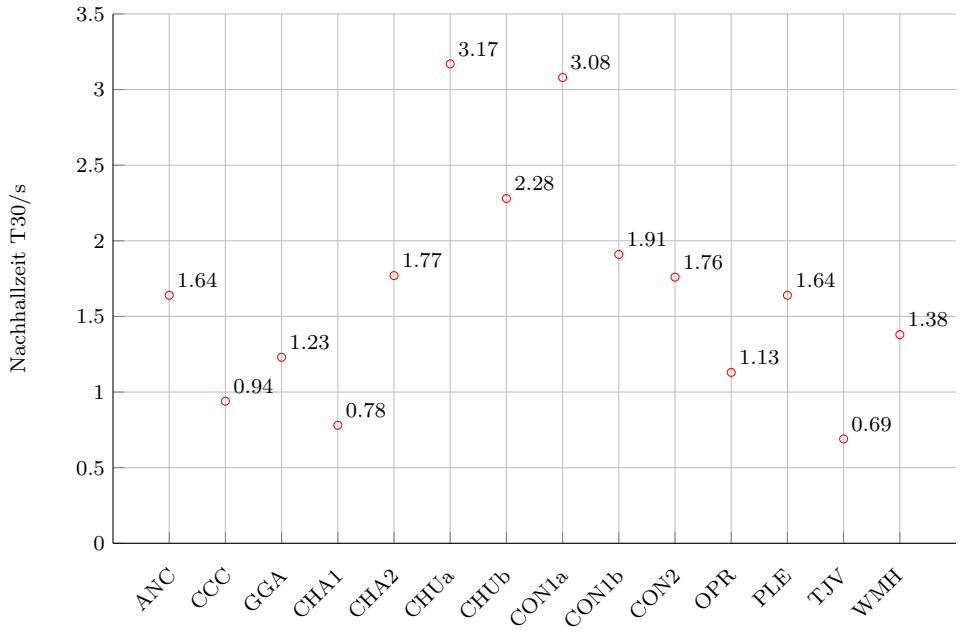


Abbildung 3.2.: Streuung der mittleren Nachhallzeiten T30 in s für alle erstellten Modelle.

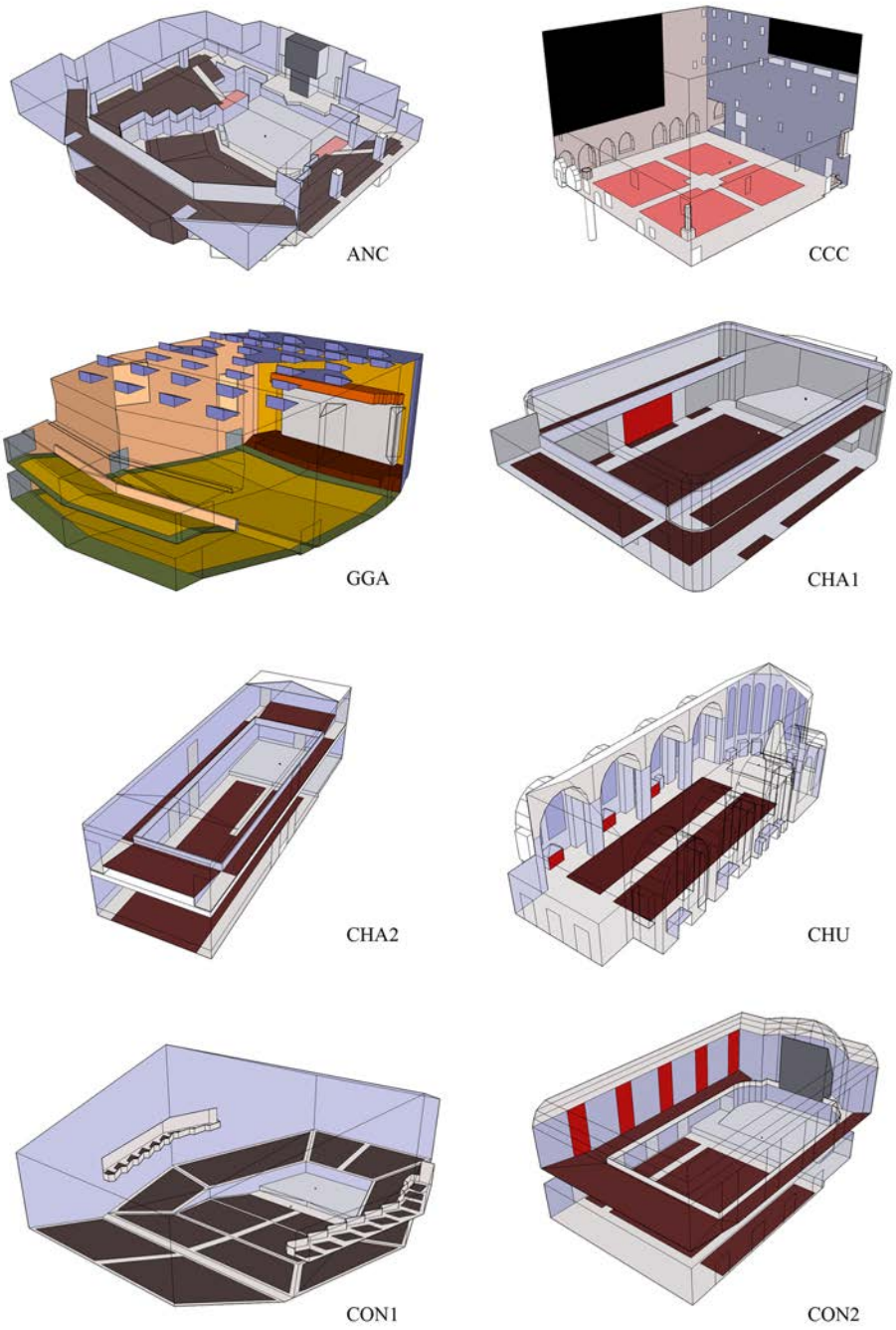


Abbildung 3.3.: Im Versuch verwendete Modelle (1-8).

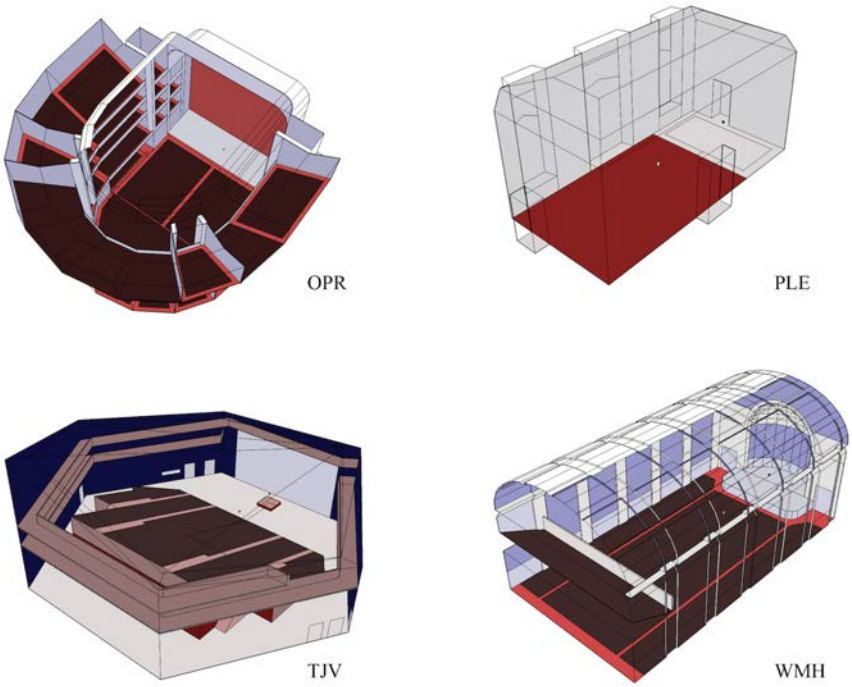


Abbildung 3.4.: Im Versuch verwendete Modelle (9-12).

3.1.4. Positionierung der Musiker

Die Positionen der Musiker für den späteren Laboraufbau müssen schon in der Simulation festgelegt werden. Die Richtungsinformation für die Schallquellen aus Sicht der Empfänger wird mit der Berechnung in die BRIRs hinein kodiert. Die Ortung der Quellen ist in diesem Versuch von entscheidender Bedeutung, da mehrere Musiker zusammen spielen und der akustische zum visuellen Reiz passen muss. Die simulierte Luftabsorption sowie die individuellen Richtcharakteristiken der Instrumente ermöglichen die Erfassung der Entfernung und Ausrichtung der Quellen durch die Musiker. Um eine plausible Auralisation zu erreichen, sollten also die Positionen sowohl im Computermodell als auch im Versuchsaufbau identisch sein und korrekte Richtcharakteristiken verwendet werden. In SketchUp wurde die übliche Anordnung eines Quartetts auf einer Bühne konstruiert. Es wurde eine Positionierung gewählt, die auch im Labor umgesetzt werden kann. Auf eine exakte Ausrichtung der Blickrichtung konnte dabei verzichtet werden, da wie in Abschnitt 3.1.6 beschrieben, die BRIRs für horizontale Winkel von -50 bis +50° berechnet wurden, was eine ausreichende Toleranz mit sich bringt.

Die verwendete Anordnung ist in Abbildung 3.5 dargestellt. Quellen werden in SketchUp durch das RAVEN-Plugin als Komponenten in Form eines Lautsprechers eingefügt, die binauralen Empfänger in Form eines Kunstkopfes. Die verwendeten Positionen für die Quellen, mit Koordinaten relativ zu den Empfängern, wurden Schärer Kalkandjiev und Weinzierl (2015) entnommen und sind zusammen mit den in dieser Arbeit verwendeten Nummern als Referenz für die Instrumente in Tabelle 3.2 aufgelistet. Die endgültige Positionierung geschah jedoch nicht im CAD-Modell, sondern in Matlab, um die Positionen dynamisch innerhalb des Skriptes bearbeiten zu können. Das dynamische Einfügen des Quartetts in Matlab bietet die Möglichkeit Quell- und Empfängerpositionen schnell für Berechnungen ändern zu können, ohne alle Modelle im CAD-Programm bearbeiten zu müssen. In jedem Modell wurde 2 m von der Bühnenkante entfernt eine Quelle auf der Bühne in 1, 2 m Höhe platziert. Diese diente im Matlab Skript als Koordinatenursprung für die Positionierung des Quartetts. Außerdem wurde jedem Modell ein Empfänger im Publikumsbereich, 4 m von der Bühnenkante entfernt, eingezeichnet. Dieser diente zur Evaluation der Quellausrichtungen während der Testphase der Simulationen.

Nr.	Instrument	x/cm	y/cm	z/cm
1	Violine	20	-20	0
2	Violine	20	-20	0
3	Viola	20	-20	0
4	Cello	0	-40	-60

Tabelle 3.2.: Koordinaten der verwendeten Instrumente relativ zum Empfänger in 1,2m Höhe. Positive x-Werte zeigen vom Empfänger aus gesehen nach links, negative y-Werte nach vorne (Schärer Kalkandjiev und Weinzierl, 2015).

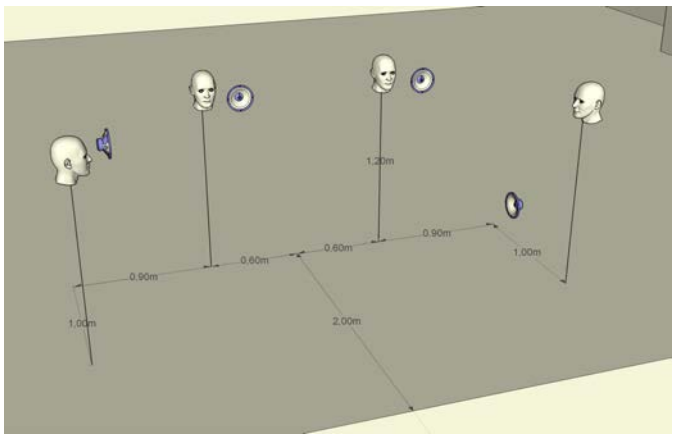


Abbildung 3.5.: Positionen der Musiker als Quellen und Empfänger im CAD-Modell. V.l.n.r: Violine (1), Violine (2), Viola (3), Cello (4).

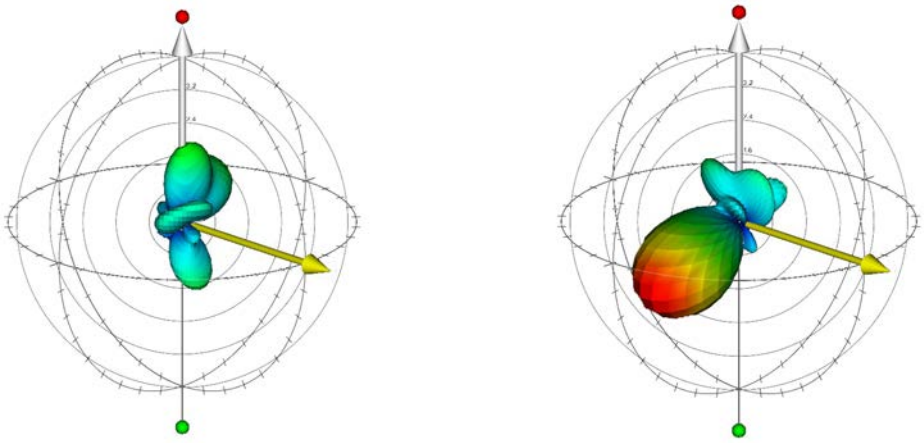
3.1.5. Richtcharakteristiken

Für alle vier simulierten Instrumente wurden passende Richtcharakteristiken verwendet. Die Daten stammen aus der von Pollow et al. (2010) erstellten Datenbank, für die Charakteristiken von einer Vielzahl an Musikinstrumenten gemessen wurden. Die benötigten Charakteristiken lagen im open Directional Audio File Format (openDAFF) vor, welches RAVEN für Richtcharakteristiken und HRTFs verwendet. Sie sind über alle Frequenzbänder gleich auf die diffus abgestrahlte Gesamtenergie normiert. Diese Normierung ist also unabhängig von der Richtung. Die Energie jedes Frequenzbandes hat, relativ zur Gesamtenergie der Quelle, einen definierten Pegel. Die Verwendung dieser Variante, im Gegensatz zu einer Normierung auf die frontale 0°-Ausrichtung, wurde gewählt, da bei einer Normierung auf einen definierten Winkel auch die Aufnahme des zu reproduzierenden Signals am Instrument aus der selben Richtung erfolgen muss. Die Richtcharakteristiken wurden, wie durch Pollow et al. (2010) beschrieben, mit einer Anordnung aus Mikrofonen in den Flächenmittelpunkten eines abgestumpften Ikosaeders[6] im Fernfeld gemessen. Im hier durchgeführten Laborversuch ist jedoch nur eine direkte Mikrofonierung nah an der Quelle möglich, weshalb eine Normierung auf eine Richtung nicht korrekt wäre. Am Punkt der Mikrofonierung nah am Instrument ist das aufgenommene Signal schon von der Richtcharakteristik des Instruments beeinflusst. Mit einer Normierung auf eine definierte Raumrichtung würde die simulierte Charakteristik die aufgenommene überlagern. Extrema in den einzelnen Frequenzbändern führen dann zu starken Verfärbungen in der Auralisation. Die Verwendung der diffus normierten Charakteristik schwächt diesen Effekt ab, da sich die Normierung auf die gesamte Energie bezieht. Dadurch ist sie für die Auralisation besser geeignet.

In Abbildung 3.6a ist die Richtcharakteristik einer Violine bei 2 kHz als Balloonplot dargestellt. Deutlich erkennbar ist, dass die Hauptabstrahlrichtung in diesem

[6]Körper mit 12 regelmäßigen Fünfecken und 20 regelmäßigen Sechsecken (ähnlich einem Fußball)

Frequenzbereich nach oben deutet. Trotz der zahlreichen Nebenkeulen ist eine maßgebliche Richtwirkung vorhanden. Noch deutlicher wird die Ausprägung der Richtwirkung im Frequenzband von 16 kHz, die in Abbildung 3.6b dargestellt ist. Die deutlich frequenzabhängige Charakteristik macht die Notwendigkeit der Implementierung von Richtcharakteristiken für eine natürliche Auralisation und korrekte Schallfeldsimulation deutlich.



(a) 2 kHz (b) 16 kHz

Abbildung 3.6.: Darstellung der Richtcharakteristik einer Violine als Balloonplot.

Auf Empfängerseite wurde der HRTF-Datensatz des FABIAN-Roboters aus Brinkmann et al. (2013) in die Simulation implementiert. Die Verwendung individueller Datensätze für jeden Musiker kam hier nicht in Frage, da die Versuchsumgebung für mehrere Musiker und wiederholte Versuche nutzbar sein soll und die individuelle Messung von HRTFs für jeden Musiker mit zu großem Aufwand verbunden ist.

3.1.6. Freiheitsgrade

Wie in den vorangegangenen Abschnitten verdeutlicht, haben die Richtcharakteristiken und Ausrichtungen der Musiker im Raum entscheidenden Einfluss auf das Schallfeld, in welchem sich die Empfänger, also die Musiker selbst befinden. In Abschnitt 2.2.3 wurde bereits beschrieben, von welcher Bedeutung auch die Bewegungen der Empfänger im Schallfeld sind. Einerseits wird durch Kopfbewegungen eine verbesserte Ortung der Quellen erreicht. Andererseits darf sich das Schallfeld durch eine Bewegung des Empfängers nicht verändern. Andernfalls würden sich die Quellen bei einer Kopfdrehung und Wiedergabe der Ohrsignale über einen Kopfhörer ebenfalls bewegen. Außerdem bewegen sich Musiker gewohnheitsgemäß beim Musizieren mit ihrem Instrument passend zur Musik oder um z. B. Blickkontakt mit den anderen Musikern im Ensemble aufzunehmen. Für die Realisierung der VAE waren also für verschiedene Blickrichtungen der Musiker Freiheitsgrade zu

bestimmen. Für jeden Winkel mussten dann einzelne BRIRs berechnet werden. Der automatisierte Ablauf dieser Simulationen wurde in Matlab implementiert.

Während Schöps et al. (1997) die maximal möglichen Freiheitsgrade für die Rotation, Ante[7]-, Retroflexion[8] und die Lateralflexion[9] bestimmt haben, widmeten sich Thurlow et al. (1967) bereits 1967 den typischen Bewegungen von Hörern. Wie in Tabelle 3.3 ersichtlich, stellte er für die Rotation Winkel von $\pm 42°$ bei hörtypischen Bewegungsabläufen fest. Die Werte für die Ante- und Retroflexion betrugen $\pm 15,2°$ und für die Lateralflexion $\pm 11,6°$. Um den hörtypischen Bewegungen von Musikern gerecht zu werden, wurden hier Rotationsgrenzen von $\pm 50°$ festgelegt. Die Winkel sind jeweils als Azimutalwinkel, also Winkel zur Meridianebene zu verstehen. Somit sind die maximal nötigen Winkel eingeschlossen und es wird ein zusätzlicher Spielraum für die exakte Positionierung der Musiker im Raum mit gewährt (siehe Abschnitt 3.1.4). Für die Ante- und Retroflexion wurden, wie im Versuch von Schärer Kalkandjiev und Weinzierl (2015), Winkel zwischen $-30°$ und $+20°$, also Vertikalwinkel zwischen Horizont und Zenit (Elevation), festgelegt, um den Musikern größere Bewegungsfreiheit beim Musizieren zu gewähren.

Richtung	hörtypisch (Thurlow et al., 1967)	maximal (Schöps et al., 1997)
Rotation	$\pm 42°$	$\pm 81,8°$
Ante-/Retrofelexion	$\pm 15,2°$	$74,7°/71,3°$
Lateralflexion	$\pm 11,6°$	$\pm 45,9°$

Tabelle 3.3.: Hörtypischer und maximaler Bewegungsbereich (Popko, 2013).

Die Lateralflexion wurde für diesen Versuchsaufbau vernachlässigt, da die zusätzlichen Berechnungen die Rechenzeiten erheblich verlängert hätten und der verwendete Software-Renderer für die Binauralsynthese hierfür keine Unterstützung bot. Popko (2013) hat gezeigt, dass auch die Kopf-über-Torso-Orientierung (HATO) Einfluss auf die HRTF hat. Jedoch würde sich die Rechenzeit mit der Anzahl der zu berechnenden HATO-Winkel ebenfalls vervielfachen. Da für jeden Raumwinkel eine eigene Berechnung nötig ist, liegt die Anzahl der Berechnungen für alle Azimutwinkel bei 101 (inklusive der 0°-Richtung). Für die Berechnung inklusive der veranschlagten Elevationswinkel ergeben sich bereits 5151 Durchläufe. Die Vervielfachung dieser Anzahl mit der Anzahl an gewünschten lateralen und Kopf-über-Torso-Ausrichtungen wäre für die gewünschte Anzahl an Räumen nicht in einem geeigneten Zeitraum durchführbar.

Um die Anzahl weiter zu verringern, kann die Winkelauflösung für den Azimut und die Elevation verkleinert werden. Lindau et al. (2008) stellten fest, dass in ihren Hörversuchen kein Proband eine kleinere Auflösung als 3° für horizontale und vertikale Quellenverschiebungen ausmachen konnte. Für diesen Versuchsaufbau

[7] Anatomie: Beugung nach vorne
[8] Anatomie: Beugung nach hinten
[9] Anatomie: Seitwärtsneigung

wurde eine Auflösung von 2° für alle Raumrichtungen gewählt, welche somit unter diesem Schwellwert liegt. Die Anzahl N der Berechnungen konnte dadurch auf $N = (\frac{100}{2} + 1) \cdot (\frac{50}{2} + 1) = 1326$ reduziert werden. Zu beachten ist jedoch, dass dies nur für eine Quellen- und Empfängerkombination gilt. Für jeden Empfänger in einer Simulation müssen alle für ihn hörbaren Quellen einzeln berechnet werden, da durch ihre unterschiedliche Position und Richtcharakteristik auch unterschiedliche Raumimpulsantworten an der Empfängerposition entstehen. Für alle daraus resultierenden 16 Kombinationen und alle 14 Räume müssen demnach 297.024 Berechnungen durchgeführt werden:

$$N = Azi \cdot Ele \cdot S \cdot R \cdot rooms \cdot HATO, \quad \text{mit} \tag{3.3}$$

$$\begin{aligned}
N &= \text{Anzahl Berechnungen} \\
Azi &= \text{Anzahl Azimuth-Winkel} \\
Ele &= \text{Anzahl Elevations-Winkel} \\
SR &= \text{Anzahl Quell- und Empfängerkombinationen} \\
rooms &= \text{Anzahl Räume} \\
HATO &= \text{Anzahl HATOs}
\end{aligned}$$

Wichtig für die Geschwindigkeit der Simulationen ist die Tatsache, dass das Ray Tracing nur einmal für jede Quell- und Empfängerkombination berechnet werden muss, da sich das diffuse Schallfeld bei einer einfachen Drehung des Empfängers um seinen Ursprung nicht ändert. RAVEN legt hierfür einen Radius fest, innerhalb dessen eine Bewegung stattfinden darf, ohne dass das Ray Tracing neu berechnet werden muss (s. Abschnitt 3.1.7). Selbstverständlich müssen aber, wie durch Schröder (2011, S. 87) beschrieben, die Faltungen mit den entsprechenden HRIRs für jeden Raumwinkel passend ausgeführt und das Spiegelschallquellenmodell neu berechnet werden.

3.1.7. BRIR Berechnung

Die Berechnung der BRIRs, mit den in den vorherigen Kapiteln beschriebenen Eigenschaften, geschah über das für die Versuchsumgebung entwickelte Matlab Skript *VSE_main.m*, aus dem heraus die Simulationen in RAVEN gestartet werden. Da vor den endgültigen Berechnungen viele Abläufe und Simulationsparameter erst getestet werden mussten, wurde das Skript weitgehend dynamisch und strukturiert mit Funktionen aufgebaut. Zudem wird eine wiederholte Berechnung mit neuen Raummodellen auf diese Weise vereinfacht.

Im Folgenden soll der Programmablauf des Skripts beschrieben werden. Der Programmablaufplan in Abbildung 3.7 zeigt dessen grundlegende Struktur. Das Skript

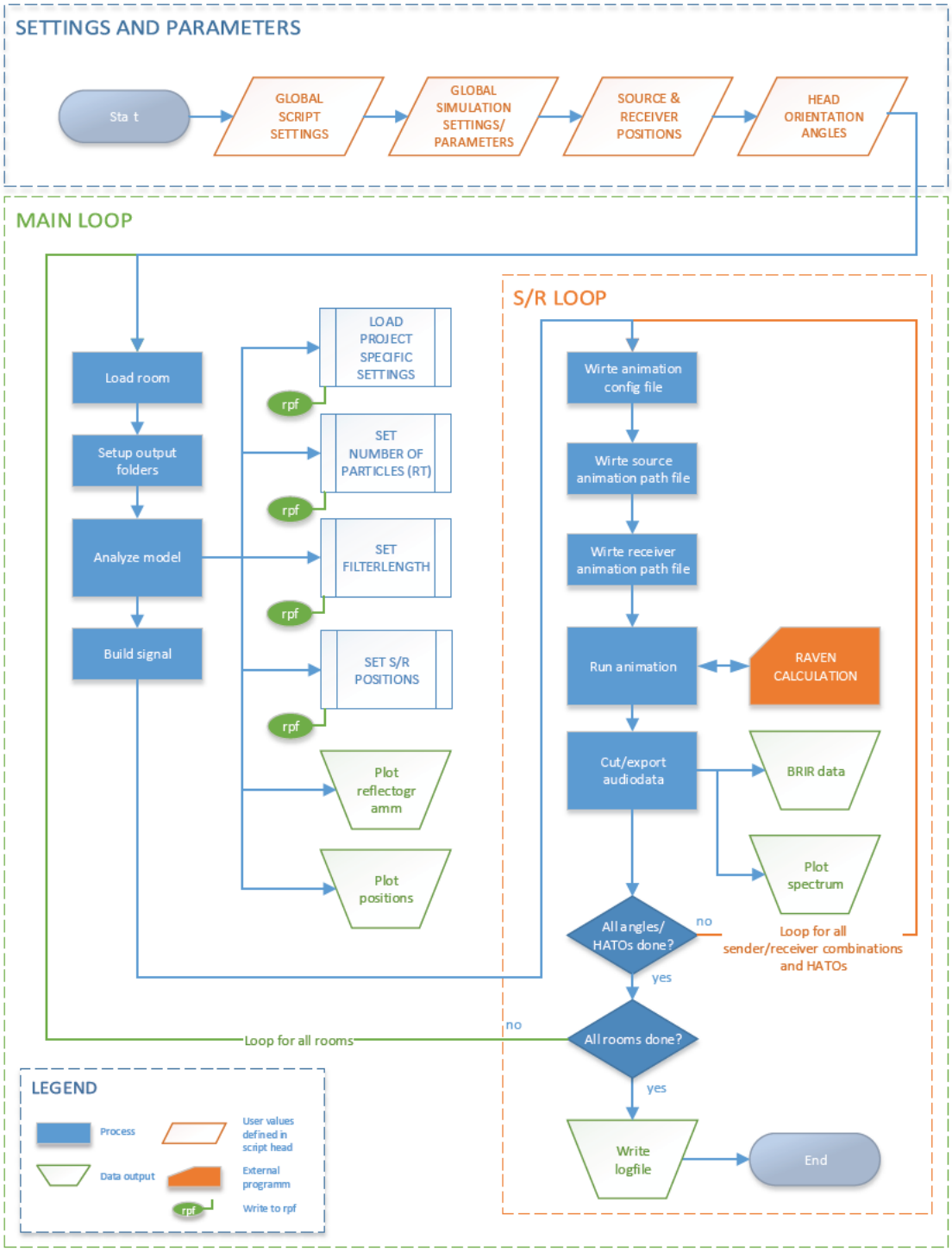


Abbildung 3.7.: Programmablaufplan des VSE Matlab-Skripts zur BRIR-Berechnung mit RAVEN.

kann in zwei Bereiche aufgeteilt werden: Es beginnt mit einem Kopf mit Einstellungen, Parametern und grundlegenden Kalkulationen. Hier bietet sich die Möglichkeit, alle relevanten Parameter für die Simulation in RAVEN festzulegen oder berechnen zu lassen. Als Parameter kommen die Positionen der Quellen und Empfänger, ihre Eigenschaften, die Freiheitsgrade für die Kopfausrichtungen und die zu exportierenden Dateiformate in Frage. Auf den Kopf folgt die eigentliche BRIR-Berechnung, die in mehreren Schleifen durchgeführt wird. Für eine effiziente Berechnung der BRIRs für alle Kopfausrichtungen wird das Animationsmodul von RAVEN verwendet.

Im Skript verwendete Funktionen sind in einzelne Dateien ausgelagert. Diese sind in Abbildung C.1 aufgelistet und auf dem Datenträger zu dieser Arbeit enthalten. Ihre Funktion und die verwendeten Argumente sind im Kommentar jeder Datei aufgeführt. Im weiteren Verlauf dieses Kapitels wird auf die besprochenen Abschnitte des Haupt-Skriptes durch Verweise auf die Überschriften im Kommentar verwiesen: (→ BEISPIEL VERWEIS). Funktionen und Parameter im Skript werden im Text *hervorgehoben*. Die Wahl der Parameter und ihre detaillierte Beschreibung enthält Abschnitt 3.1.8.

Animationsmodul

Um eine möglichst effiziente Berechnung der BRIRs zu ermöglichen, wurde das Animationsmodul von RAVEN verwendet. Hierdurch können unnötige Berechnungen des virtuellen Schallfeldes vermieden werden. Wie bereits in Abschnitt 3.1.6 beschrieben, müssen bei einer reinen Rotation des Empfängers in RAVEN nur die Spiegelschallquellen und die Faltung mit der Empfängercharakteristik für jede neue Position neu berechnet werden. Eine Neuberechnung des Ray Tracings ist nicht notwendig, da sich die Aussendung und die Schnittpunkte der Strahlen mit dem kugelförmigen Detektor nicht ändern.

Die ursprüngliche Bestimmung des Animationsmoduls liegt in der Berechnung von Auralisationen mit bewegten Empfängern und Quellen im virtuellen Raum. Hierfür werden einzeln, für eine Quelle und einen Empfänger, Textdateien angelegt, welche Bewegungspfade durch den virtuellen Raum enthalten. Bewegungsänderungen werden mit Zeitstempeln und Koordinaten in diesen Dateien festgehalten. Die Pfade können direkt in SketchUp erstellt werden, wurden jedoch zwecks Automation des Skriptes dynamisch in MATLAB erzeugt.

Für die Auralisation der VAE und der Bewegung wird RAVEN eine Audiodatei als Eingangssignal übergeben. Für jeden Punkt der Bewegung, der einen zuvor definierten Radius (*rayTracingUpdateRadius*) übersteigt, wird das Schallfeld neu berechnet und die entsprechende BRIR erzeugt. Diese wird mit dem Eingangssignal gefaltet und ergibt somit das Ausgangssignal für die Auralisation. Um reine, von

einem Quellsignal unabhängige, BRIRs zu erhalten, wurden hier als Eingangssignal für die Auralisation Dirac-Impulse verwendet.

Als Eingangssignal wird ein über den gesamten Zeitraum des Bewegungsablaufs andauerndes Signal benötigt. Es wird mit der jeweils für den Zeitpunkt und die Position erzeugten BRIR gefaltet. Für die hier geforderte Anwendung - eine Berechnung von BRIRs für jede definierte Ausrichtung des Empfängers - wird ein Eingangssignal erzeugt, welches eine Folge von Dirac-Impulsen entsprechend der Anzahl der zu berechnenden Blickrichtungen enthält. Dabei ist der Abstand der Stöße abhängig von der Nachhallzeit des zu berechnenden Raums. Um für jede Position eine BRIR berechnen zu können, muss die Antwort des Raums auf den Dirac-Stoß vollständig abgeklungen sein, bevor der nächste Impuls folgt. Dieser Zeitraum wird im Matlab-Skript mit dem Parameter *filterLength* beschrieben. Vor der Berechnung der BRIRs wird mit einer einfachen Simulation die Nachhallzeit des Modells berechnet. Die doppelte Nachhallzeit wird dann als Wert für den Parameter *filterLength* verwendet. Die doppelte Länge bewirkt definitionsgemäß einen Störabstand von 120dB des nächsten Impulses zur Raumantwort. Da die Faltung der BRIRs mit dem Eingangssignal blockweise geschieht, muss ein Parameter *blockLength* definiert werden und der Parameter *filterLength* samplegenau einem vielfachen dieser Blocklänge entsprechen:

$$filterLengthSample = 2RT \cdot Fs - (2RT \cdot Fs \bmod blockLength) \tag{3.4}$$

Für das Eingangssignal wird nun eine Folge von Nullen mit der Länge *filterLengthSample* gebildet. Das erste Sample dieser Folge erhält den Wert 0,999 als Dirac-Impuls. Entsprechend der Anzahl der zu berechnenden Blickrichtungen des Empfängers wird diese Folge wiederholt. Die Berechnung in RAVEN mit dem Animationsmodul verursacht in der Audiodatei am Ausgang eine Latenz einer halben Blocklänge (Ackermann und Ilse, 2015, S. 53). Außerdem wird die erste halbe Blocklänge an Samples nicht verarbeitet. Damit der erste Impuls trotzdem in die Simulation einfließt, werden Nullen in der Anzahl einer Blocklänge dem Signal vorangestellt. Die resultierende Sample-Folge wird als WAV-Datei dem Animationsmodul übergeben.

Passend zu diesem Signal wird nun für den Empfänger ein Bewegungspfad erstellt. Beginnend mit der Urspungsposition zum Zeitpunkt Null wird die Position im Raum und die Blickrichtung in kartesischen Koordinaten festgehalten. Diese erste Position wird für den Zeitraum der *filterLength* plus einer Blocklänge (analog zum Audiosignal) und plus einer halben Blocklänge (Latenz) minus ein Sample gehalten. Das Ende des Zeitraums wird wieder mit denselben Positionsdaten in die Textdatei geschrieben. Für die nachfolgende Positionsänderung wird der nächste Startzeitpunkt, ein Sample nach dem letzten Zeitraum bestimmt und wieder um die Länge der *filterLength* gehalten. Somit erfolgt die erste Positionsänderung/Drehung ein Sample vor dem folgenden Impuls. Für diesen Zeitraum wird die neue Position, z.B. eine Drehung

um 2° im Azimuth gespeichert. Die Funktion *VSE_writeAnimPath* (Quellcode A.2) erstellt diese Textdatei mit folgender Syntax:

```
Koordinatensystem
Time(s), Pos_x, Pos_y, Pos_z, View_x, View_y, View_z, Up_x, Up_y, Up_z
...
```

Die erste Zeile gibt dabei das verwendete Koordinatensystem anhand des Wertes *Raven* oder *SketchUp* an. Wie in Abschnitt 3.1.2 beschrieben, verwenden RAVEN und SketchUp unterschiedliche Systeme. Die Werte für die Position werden in kartesischen Koordinaten in der Einheit m angegeben. Die Ausrichtung des lokalen Koordinatensystems des Empfängers wird durch die Vektoren View und Up für die x- und z-Achse festgelegt, woraus sich die Blickrichtung des Empfängers ergibt. Hier wird also kein Winkel angegeben, sondern der Punkt für die entsprechende Ausrichtung auf dem Einheitskreis. Im Folgenden ist ein Beispiel für eine Animationspfad-Datei mit einer Drehung eines Empfängers um 2° angegeben:

```
Raven
0.0000, 7.731000, 1.700000, 5.415000, 1.000000, 0.000000, 0.000000, 0.0, 1.0, 0.0
3.7035, 7.731000, 1.700000, 5.415000, 1.000000, 0.000000, 0.000000, 0.0, 1.0, 0.0
3.7035, 7.731000, 1.700000, 5.415000, 0.999391, 0.000000, 0.034899, 0.0, 1.0, 0.0
7.3723, 7.731000, 1.700000, 5.415000, 0.999391, 0.000000, 0.034899, 0.0, 1.0, 0.0
...
```

Für die Quelle muss ebenfalls ein solcher Pfad generiert werden. Da die Quellen für die hier berechneten BRIRs jedoch nicht bewegt werden, muss nur eine Datei mit derselben Start- und Endzeit wie im Empfängerpfad und einer festen Position erstellt werden. Zusätzlich wird eine Konfigurationsdatei für das Animationsmodul benötigt, welche Angaben zu den zu verwendenden Signalen und den Animationspfaden enthält. Hinzu kommen Einstellungen für den Algorithmus für die Faltung des Signals mit den BRIRs (s. Abschnitt 3.1.8). Diese Datei mit allen benötigten Informationen für die Animation wird RAVEN übergeben:

```
[AcousticAnimation]

sourcePositionFile = C:\..\Raven\RavenOutput\VSE_FINAL\...
            ...PRJ_*\AcousticAnimation\AnimationPaths\S0#_PRJ.txt

receiverPositionFile = C:\..\Raven\RavenOutput\VSE_FINAL\...
            ...PRJ_*\AcousticAnimation\AnimationPaths\R0#_PRJ.txt

sourceSignal  = C:\..\Raven\RavenOutput\VSE_FINAL\...
            ...PRJ_*\AcousticAnimation\Sounds\diracComb_PRJ.wav

outputSignal = C:\..\Raven\RavenOutput\VSE_FINAL\...
            ...PRJ_*\AcousticAnimation\Auralization\auraStream_S0#_R0#_PRJ.wav

blockSize = 1024
overlap = 1
rayTracingUpdateRadius = 5
ambisonics = 0
```

VSE Matlab Skript: Einstellungen und Parameter

Im Kopf des Skriptes werden zunächst Einstellungen für den Ablauf des Skripts festgelegt. Diese bestimmen, welche Berechnungen ausgeführt und welche Dateiformate und Grafiken exportiert werden sollen (→ GLOBAL SCRIPT SETTINGS). Alle Parameter und Einstellungen werden im Skript durch die Initialisierung von Variablen festgelegt. Da zu Beginn dieser Arbeit noch nicht feststand, welcher Echtzeit-Renderer am Ende verwendet werden sollte, wurden hier alle Möglichkeiten offengelassen: Mit dem Skript können jeweils mithilfe eigener Funktionen für jede BRIR reine WAV-Dateien, WAV-Dateien in einer für Ordnerstruktur fWONDER, 720-kanalige WAV-Dateien für den SSR (Sound Scape Renderer) sowie SOFA-Datensätze exportiert werden. Außerdem kann ein Datensatz als Matlab-Workspace-Datei mit Metadaten zu allen Berechnungsparametern erstellt werden. Dieser kann zur Archivierung genutzt oder beliebig weiterverarbeitet werden. Tabelle 3.4 listet die möglichen Dateiformate und ihre Anwendungsgebiete auf.

Format	Daten	Dateityp	Bitrate	Kanäle	Ziel
WAV	Audiodatei	*.wav	24 Bit	2 (L/R)	SW-unabhängig
fWONDER	Audiodatei	*.wav	24 Bit	2 (L/R)	fWONDER Renderer
MAT	Matlab Datensatz	*.mat	32 Bit		archivierung, SW-unabhängig
SSR720	SSR Datensatz	*.wav	24 Bit	720 (0°L, 0°R, 1°L,1°R...)	Sound Scape Renderer
SOFA	SOFA Datensatz	*.sofa	24 Bit		Sound Scape Renderer Beta

Tabelle 3.4.: Liste der vom Skript unterstützen Export-Varianten

Der SSR unterstützt ursprünglich nur die Auralisation für horizontale Kopfausrichtungen. Für diese Arbeit wurde eine Betaversion des Renderers mit Unterstützung von SOFA-Datensätzen und den enthaltenen BRIRs für Azimuth und Elevation verwendet. Deshalb wurden hier zur Archivierung Matlab-Datensätze und für den Renderer SOFA-Dateien exportiert. Das SOFA-Format ist ein durch die Audio Engineering Society (AES) als AES69-2015 standardisiertes Dateiformat für räumliche, akustische Datensätze von HRTFs oder DRIRs[10] (Majdak et al., 2013).

Nach Lindau et al. (2012) kann der Teil einer Raumimpulsantwort für eine Auralisation, der ausschließlich stochastischen Nachhall enthält, durch einen festen, für alle Kopfausrichtungen gleichen Tail, ersetzt werden. Dies wird durch die zunehmende Diffusität und die reflexionsdichte begründet, durch welche die zeitliche und spektrale Struktur des Nachhalls im späten Teil der Impulsantwort schlechter wahrnehmbar wird. Der Zeitpunkt des Übergangs von den ersten Reflexionen zum stochastischen Nachhall wird *Mixing Time* (t_m) genannt. In Anlehnung an die Versuche von Schärer Kalkandjiev und Weinzierl (2015) und Stahnke (2013) wurde für die Mixing Time in Samples eine Zweierpotenz verwendet. Der gewählte Wert 2^{15} übertrifft die mit der Methode von Lindau et al. (2012) berechnete Mixing

[10] Directional Room Impulse Responses

Time für alle Modelle und stellt dadurch eine perzeptiv korrekte Simulation sicher. Alle BRIRs im Projekt wurden beim Export in das SOFA-Format auf diesen Wert gekürzt. Anstelle der individuellen Tails für jede Kopfausrichtung wurde ein gemeinsamer Tail innerhalb eines BRIR-Datensatzes für alle Kopfausrichtungen als separate SOFA-Datei exportiert. Hierfür wurde das gekürzte Ende der BRIR für die 0°-Ausrichtung verwendet.

Neben den Audiodaten können durch das Skript auch Grafiken erstellt werden. Zum einen kann für jede Quell- und Empfängerkombination ein Reflektogramm generiert, zum anderen können die Positionen der Musiker in einer 3D-Grafik ausgegeben werden, um eine Kontrolle dieser Positionen zu ermöglichen. Beispiele für diese Grafiken sind in Abbildung D.2 und Abbildung D.3 im Anhang zu finden.

Auf diese auf das Skript bezogenen Einstellungen folgt die Definition der Parameter für die eigentliche Berechnung (→ GLOBAL SIMULATION SETTINGS). Neben den Parametern werden hier die zu berechnenden Modelle festgelegt. Für jedes Modell muss die *AC3D*-Datei mit der Geometrie des Raums und eine RPF-Datei mit den Simulationsparametern in der in Abbildung C.2 dargestellten Ordnerstruktur vorhanden sein. Innerhalb des Skripts und in der Dateistruktur können die zu den Modellen gehörigen Dateien immer an der in Tabelle 3.1 genannten Abkürzung erkannt werden (hier dient *PRJ* als Platzhalter). Im Ordner *ORIGINAL* befindet sich die ursprüngliche RPF-Datei für jedes Modell, welche nur die Quelle für die Urspungsposition und den Publikumsempfänger enthält. Der übergeordnete Ordner *VSE_FINAL* enthält die vom Matlab-Skript manipulierte, temporäre Version der RPF-Datei. Für jede Berechnung liest das Skript die Positionsdaten aus der Originaldatei ein und schreibt diese mit den im Skript definierten Parametern in die temporäre RPF-Datei. Hierdurch kann für Testläufe eine Datei unverändert behalten werden, während eine Datei immer nur für die aktuelle Simulation gültig ist. Die weiteren Ordner *SETTINGS* und *SR* (Sender/Receiver) enthalten die projektspezifischen Parameter und die Sender- und Empfängerpositionen, die über das Skript in MAT-Dateien gespeichert und geladen werden können.

Nach der Modellauswahl und der Parameterdefinition werden die Quell- und Empfängerpositionen durch den Benutzer festgelegt (→ SOURCE AND RECEIVER POSITIONS). Hier hat der Nutzer zusätzlich die Möglichkeit auszuwählen, welche dieser Quellen und Empfänger in die Berechnung mit eingeschlossen werden sollen. Ebenso sind die zu berechnenden Freiheitsgrade (Abschnitt 3.1.6) festzulegen. Es können beliebige Winkelbereiche und -auflösungen für Azimuth und Elevation sowie optional HATO-Winkel definiert werden. Durch diese individuellen Einstellungen können gezielte Simulationen für einzelne Räume, Quellen, Empfänger und Winkelbereiche durchgeführt werden, ohne das ganze Projekt berechnen zu müssen.

VSE Matlab Skript: Berechnung

In der Schleife für die Berechnung der BRIRs sind keine Nutzereingaben mehr vorgesehen. Sie ist in zwei Hauptschleifen unterteilt: Der gesamte Teil wird für jedes zu simulierende Projekt wiederholt (→ MAIN LOOP). Der → S/R LOOP wiederholt sich für alle definierten Quell- und Empfängerkombinationen. Sofern auch HATOs berechnet werden sollen, muss auch hierfür diese Schleife für jede Kopf-über-Torso-Orientierung wiederholt werden, da für jede HATO BRIRs für alle Azimuth- und Elevations-Ausrichtungen berechnet werden müssen. Die Konsequenz für die Rechenzeit wurde in Abschnitt 3.1.6 erläutert.

Zu Beginn der Hauptschleife wird das jeweilige Projekt geladen. Das RAVEN-Framework öffnet hierzu die RPF-Datei des Projekts und lädt ihre Daten als RAVEN-Project-Objekt in den Matlab-Workspace. Dieses Objekt wird von RAVEN bereitgestellt und übernimmt alle Parameter der RPF-Datei in Form von Matlab-Variablen. Außerdem wird für das geladene Projekt eine Ordnerstruktur für die zu generierenden Dateien im Ordner *RavenOutput* erstellt (Abbildung C.3).

Das innerhalb der Hauptschleife gestartete Skript *VSE_analyzeModel.m* enthält ausgelagerte Berechnungen mithilfe derer Parameter anhand der zuvor definierten Einstellungen ermittelt werden: Als erstes wird die für die Simulation benötigte Partikelanzahl (*NumParticles*) für den Ray-Tracing-Algorithmus berechnet (Abschnitt 3.1.8). Mit diesem gewonnen Wert kann dann, wie in Abschnitt 3.1.7 beschrieben, die benötigte Filterlänge (*filterLength*) bestimmt werden. Diese anfänglichen Berechnungen geschehen mit den Quell- und Empfängerpositionen aus der originalen RPF-Datei (eine Quelle befindet sich auf der Bühne, ein Empfänger im Publikumsbereich).

Nach der Parameterberechnung wird im Skript die Position der Quelle aus dieser RPF-Datei bestimmt. Die Position dient, wie bereits erwähnt, als Ursprung für die Anordnung der zu simulierenden Quellen und Empfänger. Anhand dieses Ursprungs und den zuvor im Skript gewählten Positionen kann die absolute Position für die einzelnen Quellen und Empfänger im Modell berechnet werden.

Alle bis hierhin festgelegten oder berechneten Parameter und Positionen, die in die Berechnung von RAVEN fließen, werden mit den durch das RAVEN-Framework bereitgestellten Matlab-Funktionen in das RAVEN-Project-Objekt und gleichzeitig in die temporäre RPF-Datei geschrieben. Dies geschieht für alle Parameter im Skript *VSE_analyzeModel.m*. Die Funktionen des Frameworks sind in der Klasse RavenProject enthalten. Diese ist in der Datei *RavenProject.m* im Matlab-Ordner des RAVEN-Hauptverzeichnisses (s. Anhang C.1) enthalten.

An dieser Stelle im Skript werden das Signal und die Animationspfade wie in Abschnitt 3.1.7 beschrieben erstellt. Die Konfigurationsdatei wird zusammen mit dem RAVEN-Objekt an die Software RAVEN über einen Kommandozeilenbefehl übergeben, womit die eigentliche Berechnung gestartet wird. Das berechnete

Ausgangssignal wird in eine WAV-Datei geschrieben. Es beinhaltet die Folge der Dirac-Impulse, die mit den simulierten BRIRs gefaltetet wurden. Das Matlab-Skript lädt diese Datei und zerteilt sie wieder in einzelne Abschnitte für jede Impulsantwort, wodurch für jeden Raumwinkel eine BRIR im Matlab-Workspace abgelegt wird. Diese Impulsantworten können nun in die einzelnen in Tabelle 3.4 beschriebenen Formate exportiert werden. Dabei wird die in Abbildung C.3 dargestellte Ordnerstruktur für die Ablage verwendet.

Für jeden Durchlauf des Skriptes wird eine Protokolldatei angelegt, welche alle verwendeten Parameter und Einstellungen dokumentiert. Somit ist eine spätere Fehlersuche und ein Nachvollziehen der getätigten Berechnungen möglich. Da RAVEN selbst keine Logdatei schreibt, sondern Fehler und Warnungen nur in der Matlab-Konsole angezeigt werden, wird auch die Historie der Konsole für jede Berechnung in einer Textdatei abgelegt.

3.1.8. RAVEN Parameter

Wie im letzten Abschnitt beschrieben, kann im Skript eine Vielzahl an Parametern für die Simulation festgelegt oder berechnet werden. Diese sollen hier erläutert werden.

Direktschall

rpf.setSkipDirectSound(skipDirectSound); skipDirectSound = 1

Die Versuchsumgebung wird im reflexionsarmen Raum aufgebaut. Dadurch wird durch die Instrumente ein Schallfeld mit Direktschall ohne Reflexionen und Diffusschall erzeugt. Der Diffusschall der VAE wird durch die Simulation berechnet und über die Kopfhörer wiedergegeben. Aus diesem Grund wird der Direktschall aus der Berechnung der BRIRs ausgeschlossen. RAVENs Parameter *skipDirectSound* wird hierfür auf den Wert 1 gesetzt.

Klimawerte

rpf.setTemperature(temp); temp = 20[°]
rpf.setHumidity(humidity); humidity = 50[%]
rpf.setPressure(pressure); pressure = 101300[Pa]

Die Temperatur, der Luftdruck und die Luftfeuchte haben Einfluss auf die Schallgeschwindigkeit und die Luftschalldämpfung. Sie fließt, wie in Abschnitt 2.2.1 beschrieben, mit in die Berechnung der BRIR mit ein. Da eine exakte Einstellung auf den im Labor auftretenden Wert im Vorhinein nicht möglich ist, wurde

die Normaltemperatur von 20°, eine mittlere Luftfeuchtigkeit von 50% und der Normaldruck auf Meereshöhe von 1013hPa gewählt.

Spiegelschallquellen

rpf.setSimulationTypeIS(simTypeIS); simTypeIS = 1
rpf.setISOrder_PS(ISOrder_PS); ISOrder_PS = 3
rpf.setISOrder_SS(ISOrder_SS); ISOrder_SS = 0

Für die Berechnung der frühen Reflexionen wurde, wie in Abschnitt 2.3 beschrieben, die Berechnung über das Spiegelschallquellenmodell durchgeführt. Der Parameter *simTypeIS* schaltet diese für die Simulation ein. *ISOrder_PS* gibt die Ordnung für die Spiegelquellen an. Diese wurde auf den Wert 3 festgelegt, wodurch ein fließender Übergang von frühen Reflexionen zum, durch das Ray Tracing berechneten, Diffusschall erzielt wurde (siehe Abbildung D.2). Der Beugungsalgorithmus wurde nicht aktiviert, da er in der verwendeten Version von RAVEN noch nicht fertig implementiert war.

Ray Tracing Optionen

rpf.setSimulationTypeRT(simTypeRT); simTypeRT = 1
rpf.setTimeSlotLength(TimeSlotLength); TimeSlotLength = 1
rpf.setRadiusDetectionSphere(radiusDetector); radiusDetector = 0,5
rpf.setEnergyLoss(energyLoss); energyLoss = 63
rpf.setNumParticles(NumParticles);

Für den Diffusschall wird in RAVEN die Ray Tracing-Methode verwendet. Sie wird über den Parameter *simTypeRT* eingeschaltet. Als weitere Option kann die in Abschnitt 2.3.2 beschriebene zeitliche Auflösung *TimeSlotLength* in ms und der Radius des kugelförmigen Detektors *radiusDetector* festgelegt werden. Außerdem ist die Angabe eines Grenzwertes *energyLoss* für den Energieverlust eines Strahls möglich, bei dem seine weitere Verfolgung abgebrochen wird. Die für jedes einzelne Modell benötigte Anzahl an Strahlen bzw. Partikeln *NumParticles* wurde individuell berechnet. Hierfür stellt RAVEN die Funktion *rpf.getNumberOfParticlesRecommendation* zur Verfügung. Die benötigte Anzahl wird abhängig von einer gewählten Standardabweichung $\sigma_{ETC} = 1$ dB des Stärkemaßes (hier auf 1dB als JND[11] festgelegt), dem Radius des Kugeldetektors $r_d = 0,5$ m, der Schallgeschwindigkeit $c = 340\ \frac{\mathrm{m}}{\mathrm{s}}$, dem Raumvolumen V in m^3 und der zeitlichen Auflösung $\Delta t = 1$ ms der Berechnung, bestimmt:

$$N = \frac{V}{\frac{\sigma_{ETC}}{4{,}34}^2 \cdot \pi \cdot r_d^2 \cdot c \cdot \frac{\Delta t}{1000}} \tag{3.5}$$

[11] Just Noticeable Difference

Beschleunigungsalgorithmus für die Schnittpunktberechnung

rpf.setAccelerationType(accelerationType); accelerationType = 0

Für die Berechnung der Schnittpunkte von Strahlen mit Oberflächen wird ein Beschleunigungsalgorithmus verwendet, um räumliche Datenstrukturen zu durchsuchen. Neben einer einfachen Brute Force Methode (BF) kommt Spacial Hashing (SH) und Binary Space Partitioning (BSP) in Frage. BSP ist SH laut Schröder (2011, S. 101) in der Rechengeschwindigkeit deutlich überlegen. Es wird in RAVEN für die Echtzeit und Offlinesimulation verwendet, sofern keine Raumgeometrie verändert wird. Nur im Falle von solchen Veränderungen kann Spatial Hashing bessere Ergebnisse für die benötigte Rechenzeit erzielen. Für die hier getätigten Simulationen wird die BSP Methode verwendet, welche durch den Wert 0 gewählt wird.

Filtereinstellungen

rpf.setFilterResolution(filterRes); filterRes = '3rd'
rpf.setAmbisonicsOrder(ambisonicsOrder); ambisonicsOrder = 0
rpf.setFilterLength(filterLength); filterLength = 2 · T30

Die Filterauflösung gibt die Auflösung im Frequenzbereich für die Simulationen in RAVEN an. Hier kann zwischen einer Aufteilung in Oktaven oder Terzen gewählt werden. Für die bestmöglichen Ergebnisse wurde die Terzauflösung gewählt. Die Filterlänge *filterLength* gibt die Zeitdauer an, für die eine BRIR durch RAVEN berechnet wird. Sie sollte länger als die erwartete Nachhallzeit und länger als die benötigte Dauer für das Abklingen der Raumantwort um einen angestrebten Pegel sein. Bei Anwendung des Animationsmoduls wird die *filterLength*, wie in Abschnitt 3.1.7 beschrieben, als Abstand für zwei aufeinanderfolgende Impulse des Anregungssignals verwendet und aus der doppelten Nachhallzeit T30 berechnet.

Die Wahl einer Ambisonics Ordnung ermöglicht das Erzeugen eines B-Format-Signals anstelle einer binauralen Impulsantwort. Dies ist für dieses Projekt nicht nicht gewünscht, weshalb der Parameter auf 0 gesetzt wurde.

Festsetzen der Zufallsprozesse

rpf.setFixPoissonSequence(fixPoisson); fixPoisson = 1
rpf.setPoissonSequenceNumber(seqPoisson); seqPoisson = 1
rpf.setFixReflectionPattern(fixReflection); fixReflection = 1

Wie in Abschnitt 2.3.2 dargestellt, verwendet RAVEN verschiedene Zufallsprozesse für die Simulation. Das Festsetzen der angewendeten Sequenzen aus der Poisson-Verteilung auf den Wert *fixPoisson = 1* ermöglicht den exakten Vergleich zwischen

zwei aufeinanderfolgenden Berechnungen. Es werden weiterhin Zufallsprozesse verwendet, diese sind jedoch für jede Berechnung identisch, da immer derselbe Abschnitt der Poisson-Verteilung genutzt wird. Bei der wiederholten Simulation ohne die Änderung von Parametern ergibt sich damit dieselbe Raumimpulsantwort. Zu diesem Zweck wird auch das Reflexionsmuster, welches für die zufällige Streuung der Schallstrahlen an Oberflächen eingesetzt wird, auf den zufälligen, aber festen Wert *fixReflection = 1* eingestellt.

Animationsparameter

blockSize = 1024;
overlap = 1;
rayTracingUpdateRadius = 5;
ambisonics = 0;

Die Animationsparameter werden, wie in Abschnitt 3.1.7 erläutert, in die Konfigurationsdatei für das Animationsmodul geschrieben. Dabei kann die Blockgröße *blockSize* für den Faltungsalgorithmus sowie die Überlappung der Blöcke *overlap* in dieser Prozedur festgelegt werden. Als Länge wurden hier 1024 Samples gewählt. Mit dem Wert 1 findet keine Überlappung statt. Außerdem bestimmt der *rayTracingUpdateRadius* den Radius, innerhalb dessen sich ein Empfänger oder eine Quelle bewegen darf, ohne dass das Schallfeld mit dem Ray Tracing-Verfahren neu berechnet werden muss. Er definiert also eine örtliche Auflösung des Ray Tracing-Schallfeldes. Die Ambisonics-Berechnung wird, wie im letzten Absatz erwähnt, ausgeschaltet.

3.2. Versuchsaufbau

Im reflexionsarmen Raum als Labor sollte eine Versuchsumgebung aufgebaut werden, die zur Auralisation der virtuellen akustischen Umgebungen für vier Musiker mithilfe der zuvor generierten BRIRs genutzt werden kann. In dieser Laborumgebung sollten sie ohne direkte Einflüsse der verwendeten Technik im virtuellen akustischen Raum musizieren können, um ihre Reaktion auf die von Raummodell zu Raummodell veränderlichen akustischen Eigenschaften der VAE unbeeinflusst beobachten zu können. Die wichtigsten Glieder der Aufnahme- und Wiedergabekette des benötigten technischen Aufbaus sind die Mikrofonierung, Signalübertragung, Echtzeit-Faltung der Signale mit den BRIRs und die Wiedergabe über die Kopfhörer. Dazu kommt das Tracking der Kopfbewegung eines jeden Musikers für die richtige Wahl der zur Ausrichtung passenden BRIR.

Wie bereits beschrieben, dient die Wiedergabe des Schallfeldes über die Kopfhörer dazu, dem Schallfeld am Ohr des jeweiligen Musikers, die zum simulierten Raum

passenden frühen Reflexionen und den Diffusschallanteil hinzuzufügen. Der Direktschall erreicht die Musiker von den Instrumenten selbst. Daraus ergibt sich das Problem, den Pegel des wiedergegebenen Diffusschalls an den Direktschall der Instrumente anzupassen, um ein natürliches Verhältnis an der Hörposition sicher zu stellen. Hierfür wurde eine Methode von Stahnke (2013) verwendet, mit der die Pegel aus der Simulation und Realität verglichen werden. Diese Methode wurde noch um weitere Maßnamen zur Kalibrierung der Wiedergabesysteme erweitert.

Im idealen Fall einer mit der Realität identischen VAE dürfen die Übertragungspfade im System der VAE keinen Einfluss auf das Signal haben. Die Raumantwort auf das Instrumentensignal soll den Nutzer unverfälscht erreichen. Deshalb wurden sowohl die Frequenzgänge der Mikrofone als auch die der Kopfhörer, die durch ihre individuellen Eigenschaften Einfluss auf die Signalübertragung nehmen, durch eine vorherige Messung erfasst und dann in der Signalverarbeitung kompensiert.

Das Versuchsdesign der mit dem hier vorgestellten System geplanten Hörversuche sieht vor, zwischen mehreren Räumen ohne Verzögerung vergleichen zu können. Außerdem muss eine einfache Ausführung von wiederkehrenden Arbeitsabläufen mit der Softwareumgebung möglich sein. Zu diesem Zweck wurden alle Softwareaufrufe und Befehle in Bash-Skripten implementiert, die Software- und Parameteraufrufe beschleunigen.

Die für den Betrieb des Systems benötigten Skripte, Konfigurationsdateien für Soft- und Hardware, Audiodateien, Dokumentationen und die BRIR-Datensätze wurden in einer geschlossenen Ordnerstruktur mit dem Titel *VSE_System* (Abbildung C.4) organisiert.

3.2.1. Technischer Aufbau

Das hier entwickelte System zur Auralisation der VAEs mit den zuvor berechneten BRIRs wurde im reflexionsarmen Raum (RAR) aufgebaut. Dabei wurden im RAR nur die benötigten Geräte zur Aufnahme und Wiedergabe installiert, die Computer und Soundkarten wurden im Kontrollraum außerhalb des RARs in Betrieb genommen. Auf diese Weise können die Probanden ungestört vom Versuchsleiter musizieren und es kommt zu keiner Beeinträchtigung durch Lärmquellen wie zum Beispiel die Computer.

Die Aufnahme- und Wiedergabekette bestand aus einem Mikrofon, befestigt am jeweiligen Instrument, einer externen Soundkarte zur AD/DA-Wandlung, einem Computer zur Faltung des Signals mit den BRIR-Datensätzen in Echtzeit, einem Kopfhörerverstärker und extra-auralen Kopfhörern zur Wiedergabe. Die Signalkette ist in Abbildung 3.8 dargestellt. Dieses Schaubild verdeutlicht den Signalpfad für eine Quelle und einen Empfänger. Für alle vier Musiker mussten zwei unabhängige Systeme mit zwei Soundkarten und zwei Computern eingesetzt werden, um die benötigten Rechenkapazitäten bereitstellen zu können. Jeder Computer war für

die Speisung zweier Empfänger mit zwei Kopfhörern und passenden Verstärkern zuständig. Die vier benötigten Mikrofonsignale wurden durch die Soundkarte verstärkt und dann digital an das zweite System weitergeleitet. Beide Systeme wurden in allen Bereichen so ausgeführt, dass sie auch alleine und unabhängig voneinander eingesetzt werden können. Dadurch wird das Gesamtsystem mit entsprechender Hardware auf eine theoretisch beliebige Anzahl an Empfängern erweiterbar. Das System für vier Musiker mit allen Signalverbindungen ist in Abbildung 3.9 dargestellt. Anhang E beinhaltet die Schaltpläne in tabellarischer Form für den dargestellten Aufbau.

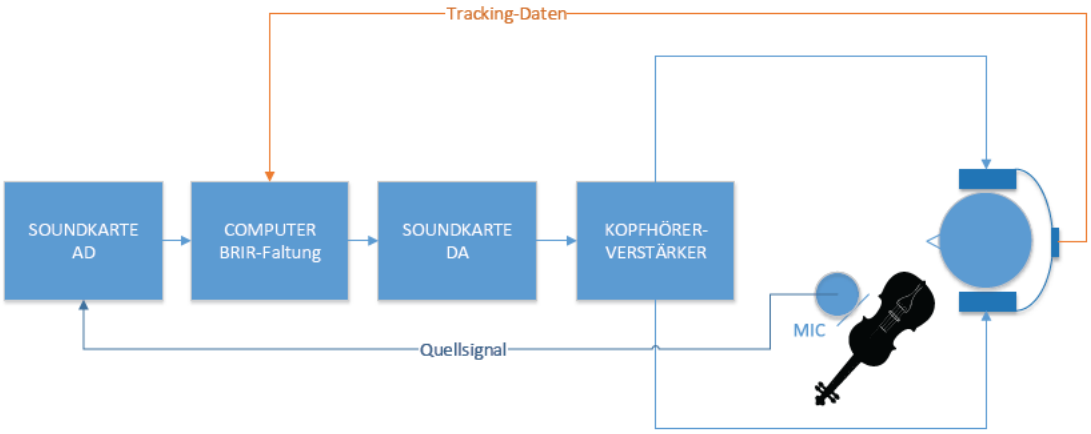


Abbildung 3.8.: Aufnahme-/Wiedergabekette des Versuchsaufbaus im RAR.

Als Soundkarten kamen eine RME Fireface Karte und eine Focusrite Scarlett 18i20 zum Einsatz. Beide Soundkarten werden per USB mit dem Computer verbunden. Bei Verwendung beider Systeme wurde die RME Karte als Master und die Focusrite Karte als Slave betrachtet. Untereinander wurden beide Karten über ADAT verbunden, um die Mikrofonsignale zu teilen. Auf diese Weise wurde kein analoger Split der vier Signale für beide Computer benötigt. Die RME Karte wurde als Master für die Verstärkung und Wandlung der Signale eingesetzt und leitete die unbearbeiteten Signale direkt an die Focusrite Karte weiter. Die sehr geringe Latenz dieser Operation (< 3 ms) bot so die technisch beste Möglichkeit die Signale für beide Computer bereitzustellen. Werden die Systeme im Standalone-Modus verwendet, muss für die Karten eine andere Hardwarekonfiguration, verwendet werden. Die Mikrofonsignale werden dann direkt mit der entsprechenden Soundkarte verstärkt und A/D-gewandelt. Die benötigten Konfigurationsdateien sind in der VSE-Dateistruktur enthalten. Über die erstellten Bash-Skripte kann definiert werden, welche Soundkarte der gerade betriebene PC verwendet und in welchem Modus (Standalone/Master/Slave).

In dem aufgezeigten System finden verschiedene Signaloperationen statt, die schematisch in Abbildung 3.10 dargestellt sind. Das Quellsignal wird mithilfe der Mikrofonvorverstärker ausgesteuert. Durch den Rendering-Computer findet die

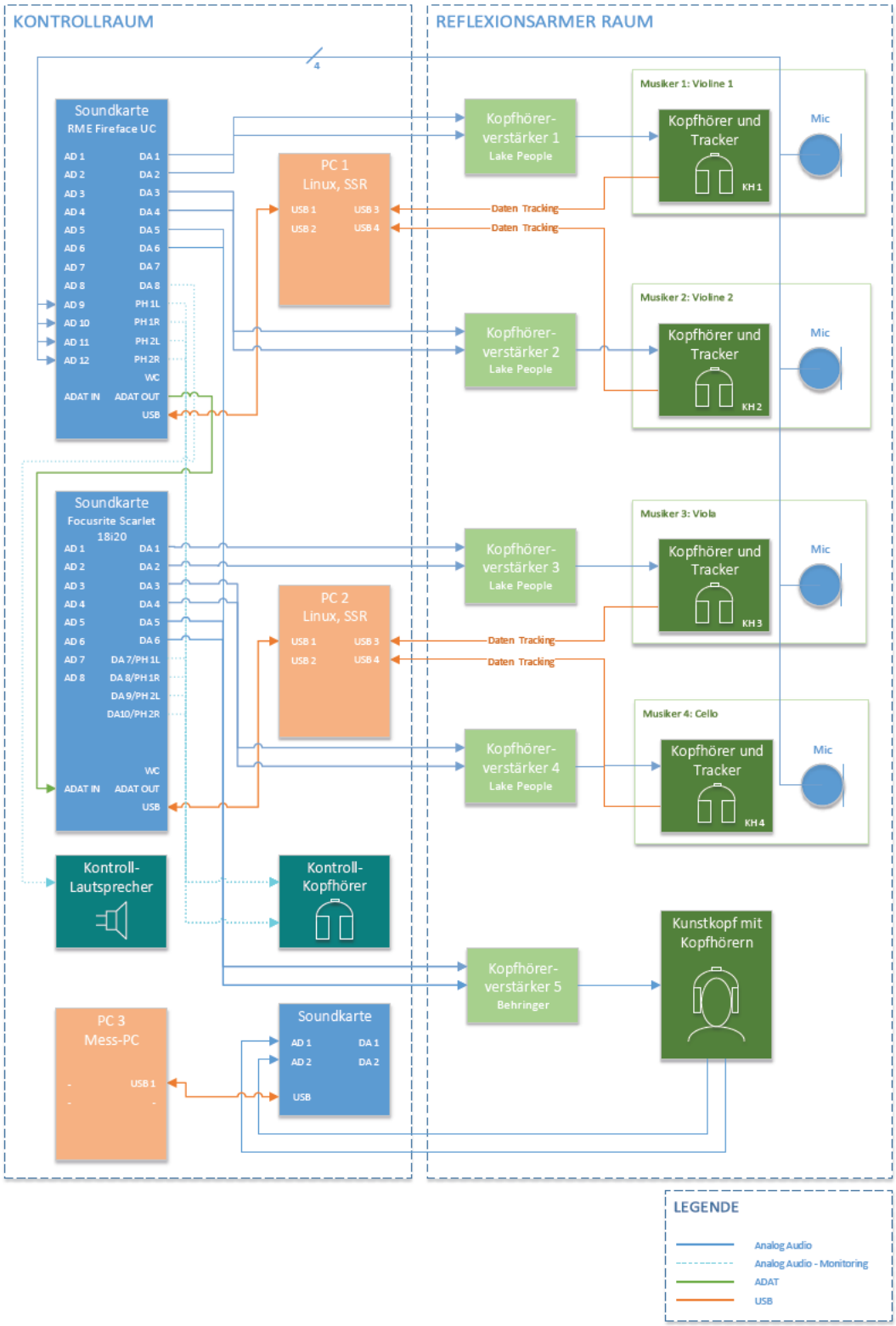


Abbildung 3.9.: Technischer Versuchsaufbau mit allen Signalverbindungen.

digitale Verarbeitung des Signals in Echtzeit statt. Zunächst wird der Pegel des monauralen Signals korrigiert (s. Abschnitt 3.2.6) und dann mit der Impulsantwort zur Mikrofonentzerrung gefaltet. Das auf diese Weise von den Einflüssen des Mikrofonfrequenzgangs befreite Signal wird mit der Raumimpulsantwort aus dem zuvor erstellten BRIR-Datensatz gefaltet. Die benötigte BRIR wird anhand der Tracking-Daten des Headtrackers auf dem Kopfhörer des Musikers ausgewählt. Das nach der Faltung des Quellsignals mit der binauralen Raumimpulsantwort nun für beide Ohren getrennt vorhandene binaurale Signal, wird mit den Impulsantworten zur Entzerrung des Kopfhörers gefaltet, um ihren Frequenzgang zu kompensieren. Auf eine erneute Pegelkorrektur (s. Abschnitt 3.2.6) für jeden Kanal einzeln folgt die Wiedergabe über die Kopfhörer.

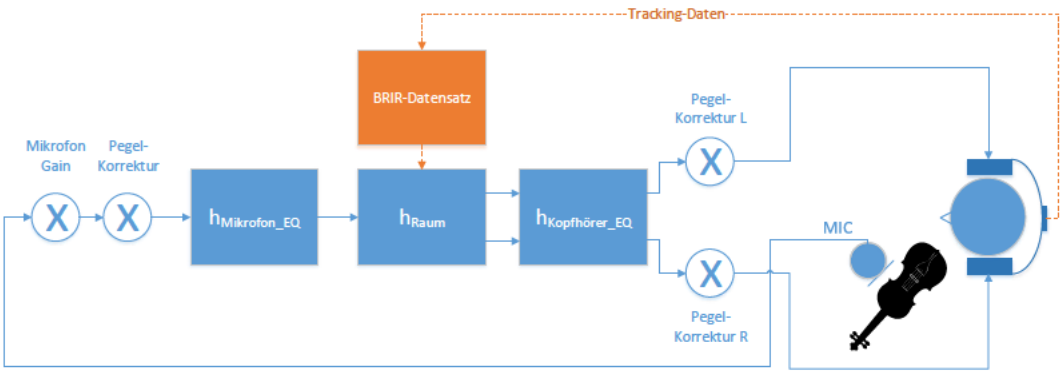


Abbildung 3.10.: Signalverarbeitung des Rendering-Computers im VSE-System.

3.2.2. Aufnahme

Das Mikrofon zur Aufnahme des Quellsignals wurde direkt am Instrument in etwa 15 cm Abstand befestigt. Um Schwankungen in der Lautstärke und der Klangfarbe der aufgenommenen Signale zu vermeiden und weil das Instrumentensignal möglichst einzeln, ohne Übersprechen von den anderen Instrumenten aufgenommen werden sollte, kam nur diese Art der Nahfeld-Mikrofonierung in Frage. Die Mikrofonierung wurde bereits in Schärer Kalkandjiev und Weinzierl (2015) entwickelt. Zum Einsatz kamen Miniatur-Lavalier-Mikrofone mit omnidirektionaler Charakteristik vom Typ Sennheiser MKE 1. Diese wurden mit einer Drahthalterung an einer typischen Aufnahmeposition im charakteristischen Hauptabstrahlwinkel des Instruments befestigt (s. Abbildung 3.11).

Die verwendeten Mikrofone weisen einen relativ linearen Frequenzgang mit einer Überhöhung von etwa 6 dB zwischen 5 und 15 kHz auf. Diese Unebenheit des Frequenzgangs wurde in der Signalverarbeitung kompensiert. Zu diesem Zweck wurde der Frequenzgang des Mikrofons im RAR mithilfe der Substitutionsmethode bestimmt: Auf einem Stativ wurde ein aktiver Nahfeldmonitor und in 2,4 m

Abbildung 3.11.: Mikrofonierung und Kopfhörer im Vorversuch mit einer Bratsche.

Abstand ein Normalmikrofon auf einem Stativ aufgestellt (s. Abbildung F.1). Mit diesem Mikrofon wurde der Frequenzgang des Lautsprechers bestimmt. In einer zweiten Messung wurde das Normalmikrofon gegen das DUT[12] an derselben Position eingetauscht. Bei der wiederholten Messung des Frequenzgangs wurde der Lautsprecherfrequenzgang invertiert zur Kompensation desselben eingesetzt und somit der Frequenzgang des DUT bestimmt. Nach einer Glättung und Invertierung des gemessenen Frequenzgangs wurde dieser in den Zeitbereich überführt. Die so berechnete Impulsantwort diente zur Kompensation des Mikrofonfrequenzgangs durch Faltung mit dem Quellsignal.

3.2.3. Signalverarbeitung

Die Signalverarbeitung in Echtzeit wurde auf zwei Computern durchgeführt. Jeder Computer war für die Ausgabe von zwei binauralen Signalen für zwei Empfänger zuständig. Neben einer schnellen Verarbeitungsgeschwindigkeit, welche durch eine hohe Taktfrequenz und schnelle Datenzugriffszeiten gewährleistet wird, war ein großer Arbeitsspeicher erforderlich, da für ein schnelles Wechseln zwischen den simulierten Räumen alle hierfür benötigten BRIRs schon im Speicher liegen mussten. Verwendet wurden zwei 64Bit PCs mit Intel 4 GHz Quadcoreprozessoren, die durch die Hyper-Threading Technologie vom Betriebssystem als acht eigenständige Prozessoren verwaltet werden. Der Arbeitsspeicher betrug 32GB DDR3 RAM.

Als Betriebssystem für die beiden Computer wurde Linux eingesetzt. Die gesamte rechnerinterne Signaldistribution wurde mit dem Soundserver JACK[13] durchgeführt. Diese Software ermöglicht ein freies Routing zwischen der eingesetzten Audiohardware und den Softwarekomponenten. Vereinfacht wird dies durch eine GUI und die

[12] Device Under Test
[13] http://jackaudio.org/

Möglichkeit Verbindungen über die Kommandozeile herzustellen und abzuspeichern. Letzteres ermöglicht die Steuerung des Routings durch Shell-Skripte. Abbildung 3.12 stellt die Softwarestruktur und den rechnerinternen Signalfluss dar.

Das unbearbeitete Quellsignal sollte rechnerintern direkt nach der A/D-Wandlung aufgezeichnet werden. Hierfür wurde ein Kommandozeilenprogramm mit dem Titel jack_capture[14] eingesetzt, welches ohne GUI auskommt und direkt aus einem Shell-Skript heraus gestartet werden kann.

Für die Entzerrung der Mikrofon- und Kopfhörerfrequenzgänge wurde die Software fCovolver[15] von Fons Adriaensen verwendet. Sie ermöglicht eine nahezu latenzfreie Faltung eines Signals innerhalb des JACK-Routings mit einer vorgegebenen Impulsantwort. Hierfür werden Konfigurationsdateien im Textformat angelegt (Quellcode B.1), welche Parameter mit den benötigten virtuellen Ein- und Ausgängen der Software und den zu verwendenden Impulsantworten im WAV-Format enthalten. Die Software wird dann über einen Kommandozeilenbefehl gestartet.

Für die benötigten Pegelkorrekturen kam die Software JACK-Mixer[16] zum Einsatz. Sie stellt beliebig viele virtuelle Ein- und Ausgänge und eine einem realen Mischpult nachempfundene GUI bereit. Hier können Pegel für jeden Kanal einzeln und für eine Summe aller Kanäle eingestellt werden. Die Konfiguration kann ebenfalls in einer Datei im Textformat gespeichert (Quellcode B.3) und über die Kommandozeile aufgerufen werden.

Alle JACK-Verbindungen wurden mit dem Kommandozeilenprogramm JMess[17] gespeichert. Die als XML-Datei (Quellcode B.5) erstellten Konfigurationsdateien wurden ebenfalls mit dieser Software per Kommandozeilenbefehl geladen.

Die eigentliche Faltung der Signale in Echtzeit mit den binauralen Raumimpulsantworten wurde, wie zuvor beschrieben, mit der Software Sound Scape Renderer (Geier et al., 2008b; Geier und Spors, 2012) durchgeführt. Der SSR ermöglicht die Faltung beliebig vieler Quellsignale mit zugeordneten Impulsantworten. Dabei kann über JACK-Verbindungen jeder Quelle ein eigenes Signal zugeführt werden. Der SSR unterstützt verschiedene Arten der Auralisation und Syntetisierung von Schallfeldern, wie z.B. auch die Wellenfeldsynthese. Hier wurde ausschließlich der BRS[18]-Renderer verwendet, welcher für die Auralisation eines Signals, gefaltet mit zuvor gemessenen oder künstlich erzeugten BRIRs, vorgesehen ist. Als Ausgangssignal liefert der BRS-Renderer des SSR ein binaurales und somit zweikanaliges Signal für einen Empfänger. Mehrere Empfänger können durch mehrere Instanzen des Programms auf einem Computer implementiert werden. Der SSR wird mit

[14] https://github.com/kmatheussen/jack_capture
[15] http://kokkinizita.linuxaudio.org/linuxaudio/index.html
[16] http://home.gna.org/jackmixer/
[17] https://github.com/jcacerec/jmess-jack
[18] Binaural Room Synthesis

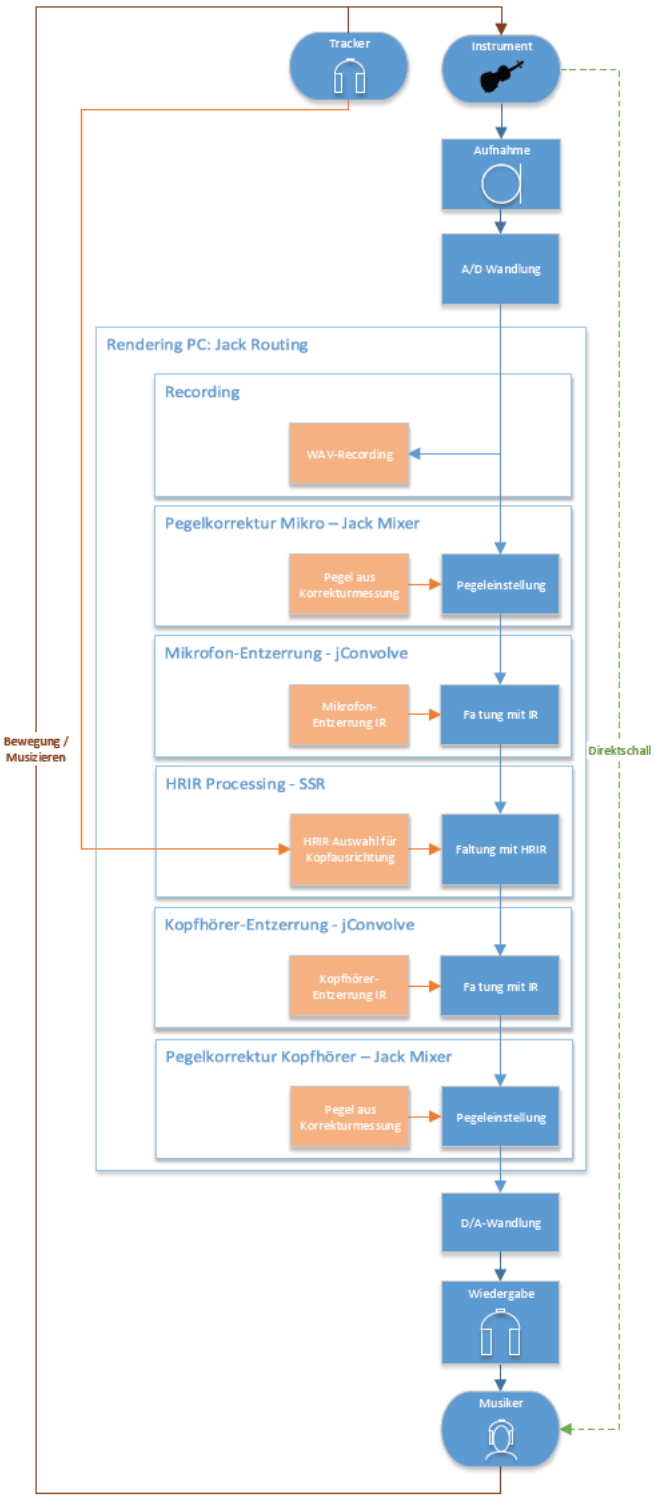


Abbildung 3.12.: Flussdiagramm Signalverarbeitung

einer ASDF[19]-Konfigurationsdatei gestartet (Geier et al., 2008a). Diese Datei im XML Format enthält Referenzen für alle Quellen, dargestellt durch einen BRIR Datensatz und einen frei definierbaren Pegel für jede Quelle. Quellcode 3.1 zeigt exemplarisch eine solche ASDF-Datei für einen Empfänger und eine Quelle mit zugehörigem Tail. Da die BRIR und der Tail in getrennten SOFA-Dateien gespeichert sind, werden sie im SSR als einzelne Quellen geladen und mit dem gleichen Signal der Quelle gespeist. Der für die Einspeisung des Signals benötigte Port für die JACK Umgebung wird hier ebenfalls angegeben. Die Position der Quelle ist nur für die visuelle Darstellung in der GUI notwendig, da dem BRS-Renderer BRIRs übergeben werden, welche die Position der Quellen schon enthalten (Geier und Spors, 2012).

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <asdf>
3     <header>
4        <name>BRS RENDERER FOR ROOM PRJ and REC 0</name>
5        <description>
6           Configuration for BRS renderer.
7           SSR instance 1, REC 0
8        </description>
9     </header>
10    <scene_setup>
11       <volume>0</volume>
12
13       <!-- PRJ -->
14
15       <source name="PRJ - REC0 - SRC0" volume="0">
16          <properties brir_file="../SOFA/PRJ/PRJ_S00_R00.sofa" />
17          <port>1</port>
18          <position x="0" y="1" fixed="true"/>
19       </source>
20
21       <source name="ANC - REC0 - SRC0 - Tail" volume="0">
22          <properties brir_file="../SOFA/PRJ/PRJ_S00_R00_tail.sofa" />
23          <port>1</port>
24          <position x="0" y="2" fixed="true"/>
25       </source>
26
27    </scene_setup>
28 </asdf>
```

Quellcode 3.1: Beispiel einer ASDF-Datei mit Parametern zur Konfiguration des SSR. Hier ist eine Quelle mit der zugehörigen Tail-Datei enthalten.

Für den Versuch wurden immer vier Räume in einer Konfigurationsdatei angelegt, die dann gleichzeitig geladen wurden. Bei vier BRIRs, vier Tails pro Empfänger und vier Räumen werden also $(4 + 4) \cdot 4 = 32$ Quellen je Sound Scape Renderer benötigt. Dies ergibt bei zwei Empfängern und somit zwei Instanzen des SSRs pro Computer $2 \cdot 32$ Quellen. Das gleichzeitige Laden der Räume ermöglicht es dem Versuchsleiter ohne Verzögerung zwischen den Räumen umzuschalten, da keine Ladezeiten für einen Neustart des SSR anfallen. Für einen Satz sehr langer BRIRs und vier Räume beträgt die Ladezeit in der verwendeten Hardwarekonfiguration

[19] Audio Scene Description Format

etwa drei Minuten. Das Umschalten zwischen den vier geladenen Räumen wird über das JACK-Routing gesteuert. Da jede Quelle im SSR einen eigenen virtuellen Audio-Eingang zugewiesen bekommt, kann das Eingangssignal einfach auf die gerade benötigten Quellen im SSR geleitet werden. Die beschriebene Konfiguration führt auf den Versuchsrechnern zu einer Auslastung der Prozessorkerne von maximal 40% im Betrieb und einer Belegung des RAMs von 75% (siehe Abbildung D.1). Eine höhere Belegung des Arbeitsspeichers führt zu einer Auslagerung der Daten auf die Festplatte durch das Betriebssystem. Dies erhöht die Zugriffszeiten auf diese Daten und ist deshalb nicht praxistauglich für das System.

Gestartet wird der SSR über die Kommandozeile. Hier werden ihm alle weiteren benötigten Parameter übergeben:

```
1 $ ssr-brs ./SSR_SETUPS/<CONFIG>.asd -v --tracker=razor
      --tracker-port=/dev/ttyUSB0 --name=BRS1 --ip-server=4712
```

Quellcode 3.2: Kommandozeilenbefehl zum Starten des SSRs mit Parametern

Der Parameter *-v* startet den SSR im Verbose-Modus, wodurch eine detailliertere Ausgabe von Warnungen und Fehlern in der Konsole eingestellt wird. Außerdem werden dem Programm der Typ des verwendeten Trackers und die verwendete Schnittstelle, ein individueller Name und eine freie *ip-server*-Adresse übergeben. Letzteres ist notwendig, wenn mehrere Instanzen des SSR betrieben werden sollen. Jede Instanz muss dann eine eigene Adresse erhalten.

Alle während eines Versuchsdurchgangs durchzuführenden Programmstarts und Befehle sind über ein Shell-Skript durchführbar. Einmal gestartet, wird der Versuchsleiter anhand eines Textmenüs in der Konsole durch einzelne Routinen geführt. Abbildung 3.13 zeigt ein Bildschirmfoto eines Rendering-PCs mit dem Kommandozeilenmenü. Das Shell-Skript ist unterteilt in das Skript *VSE_RUN.sh*, welches vom Benutzer aufgerufen wird und das Skript *VSE_Skript.sh*, das die Programmabläufe für verschiedene Aufgaben während des Versuches in Form von Funktionen enthält. Alle für den Versuch relevanten Dateien, wie die zuvor benannten Konfigurationsdateien für die Programme, die SOFA-Dateien für alle Räume, Hardware-Konfigurationen und Impulsantworten zur Kompensation sind im Ordner *VSE_System*, abgebildet in Abbildung C.4, organisiert. Hier sind auch die Skripte abgelegt, welche durch relative Pfade auf die verschiedenen Dateien zugreifen.

3.2.4. Wiedergabe

Die Wiedergabe geschah über extra-aurale Kopfhörer vom Typ AKG K1000. Diese Kopfhörer sind offen aufgebaut. Die Lautsprecher sind ohne die übliche Muschel frei vor den Ohren positioniert. Durch den Einsatz von akustisch transparenten

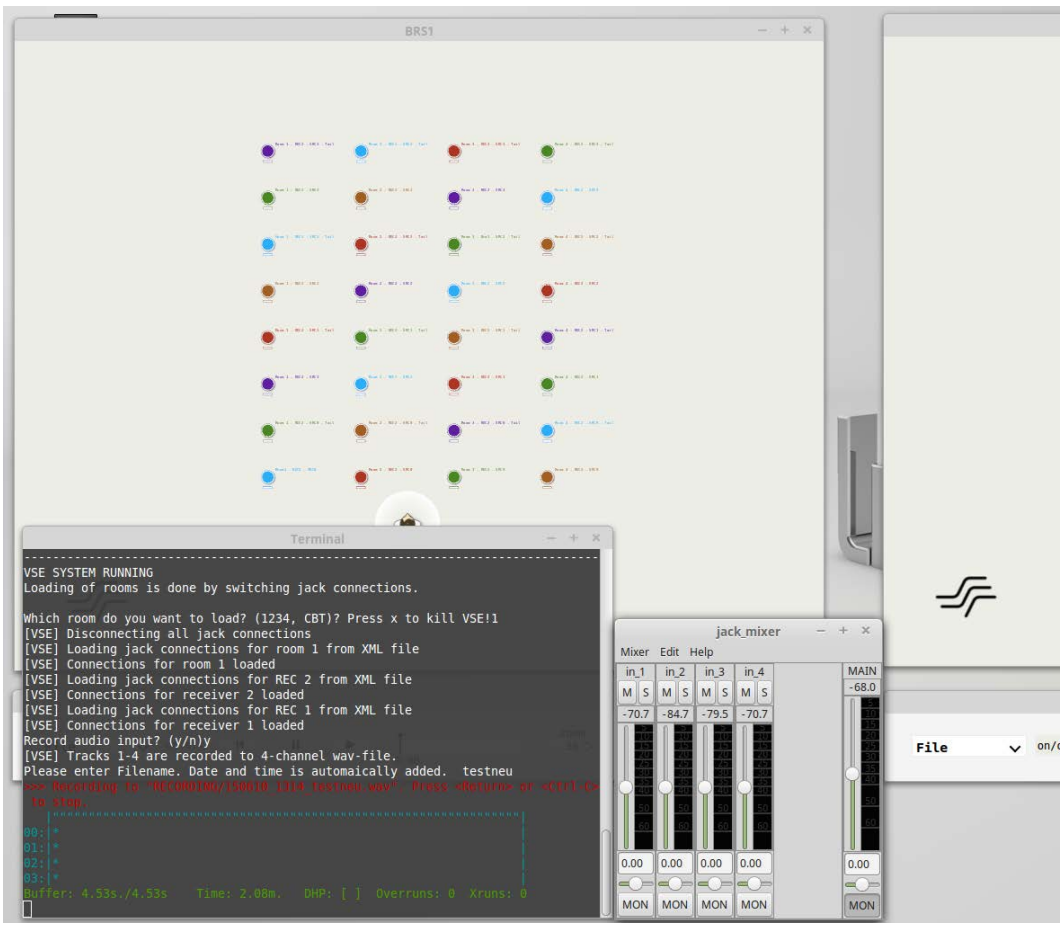


Abbildung 3.13.: Bildschirmfoto der VSE Softwareumgebung auf einem Rendering-PC. Geöffnet sind die Konsole mit dem Textmenü und einer laufenden Aufzeichnung, eine SSR-Instanz und ein Mixer zur Pegelkalibrierung.

Materialien für das Gehäuse wird der Einfluss auf das Schallfeld, in dem der Hörer sich befindet, minimiert. Dies war für diesen Versuch von besonderer Bedeutung, da die Kopfhörer den Direktschall der Instrumente nicht beeinflussen dürfen. Außerdem erfüllen sie die Anforderungen von Møller (1992) an einen Kopfhörer für die Binauralsynthese. Demnach sollte die akustische Impedanz des Kopfhörers gleich der des Schallfeldes sein. Møller et al. (1995) zeigte, dass diese Anforderung durch den K1000 erfüllt wird und er als FEC[20]-Kopfhörer betrachtet werden kann. Die Bezeichnung ist mit der Bezeichnung *offen* gleichzusetzen. Sie ist jedoch nicht mit der im kommerziellen Bereich verwendeten Bezeichnung zu verwechseln, welche lediglich Modelle mit offenem Gehäuse ohne eine Abschirmung von Außengeräuschen charakterisieren soll (Møller, 1992, S. 188).

Für alle vier benötigten Kopfhörer wurden nach der Methode von Lindau und Brinkmann (2012) Übertragungsfunktionen ermittelt und daraus Filter zur Kom-

[20]Free-air Equivalent Coupling

pensation dieser erstellt. Diese wurden, wie in Abschnitt 3.2.3 dargestellt, mit in die Signalverarbeitung integriert. Wie durch Lindau und Brinkmann (2012) beschrieben, geschah die Messung der HpTFs mit einem HATS, da auf diese Weise gewonnene Filterkurven zur Entzerrung der HpTFs zu realistischeren Ergebnissen bei der Wiedergabe führen. Als Kopfhörerverstärker wurden Verstärker des Herstellers Lake People vom Typ G109-P eingesetzt. Das Modell zeichnet sich durch für den Versuchsaufbau angemessene Geräusch- und Fremdspannungsabstände und Verzerrungswerte aus.

3.2.5. Headtracker

Zur Bestimmung der Kopfausrichtung der Musiker im Versuch wurde ein am Kopfhörer befestigter Headtracker verwendet (s. Abbildung F.3). Diese Tracker vom Typ Razor AHRS[21] basieren auf dem mit Arduino kompatiblen Board *9DOF Razor IMU* und beziehen ihre Daten aus drei Sensoren: Ein Mikrocontroller berechnet die aktuelle Ausrichtung mithilfe eines Gyroskops, eines Beschleunigungssensors und eines Magnetometers. Die zugehörige Software für die Arduino Programmierumgebung ermöglicht eine exakte Kalibrierung des Trackers, die aufgrund von Einflüssen durch das Magnetfeld des hier verwendeten Kopfhörers und der Umgebung für jeden Versuchsaufbau erneut durchzuführen ist. Die ermittelten Daten werden vom Tracker per USB an den Computer gesendet. Der Sound Scape Renderer unterstützt nativ das Protokoll des Razor Trackers.

3.2.6. Kalibrierung

In der VAE ist es für eine realistische Synthese des Schallfeldes von entscheidender Bedeutung, die Pegel der Wiedergabesysteme richtig einzustellen. Das Verhältnis zwischen dem Direktschall der Instrumente und dem künstlichen Diffusschall muss im realen Schallfeld mit dem Verhältnis in der Simulation übereinstimmen. Für ein korrektes Verhältnis muss die gesamte Wiedergabekette kalibriert werden. Dafür wurde zunächst jeder Teil der Kette für alle Empfänger auf ein einheitliches Pegelniveau gebracht und dann für jede Quelle der richtige Diffusschallanteil im Schallfeld bestimmt. Letzteres geschah mit der bereits durch Stahnke (2013) entwickelten Methode.

Die berechneten BRIRs werden, wie in Abschnitt 3.1.7 beschrieben, in RAVEN auf einem einheitlichen Pegelniveau berechnet. Die Amplituden aller Impulsantworten werden im korrekten Verhältnis zum in der Simulation ausgesendeten Direktschall berechnet. Bleibt dieser konstant, sind die Impulsantworten auch untereinander vergleichbar. Um in der VAE das richtige Verhältnis zwischen Direkt- und Diffusschall

[21] https://github.com/ptrbrtz/razor-9dof-ahrs/wiki/Tutorial

einzustellen, muss der Pegel des Direktschalls im realen Schallfeld bestimmt werden. An diesen muss der Pegel des aufgenommenen Direktschalls, der dem Renderer zugeführt wird, angepasst werden. Durch diese Anpassung wird der Diffusschallanteil durch den Renderer im richtigen Verhältnis zum zugeführten Direktschall berechnet und über das binaurale Wiedergabesystem ausgegeben.

Für diese Kalibrierung ist es zunächst wichtig, dass die Wiedergabekette vom Softwarerenderer bis zum Kopfhörer für alle Musiker einen einheitlich Pegel aufweist. Für einen passenden Test stellt das VSE-Skript eine Routine mit weißem Rauschen bereit. Durch die Verwendung von Rauschen anstelle eines Pegeltons werden Fehler in der Kalibrierung durch die unterschiedlichen Frequenzgänge der einzelnen Lautsprecher der Kopfhörer vermieden. Die Erfassung des Pegels wird dadurch über den gesamten Frequenzbereich durchgeführt und nicht auf eine einzelne Frequenz, welche genau in einer Überhöhung liegen kann, bezogen. Das Rauschen wird durch das JACK-Routing über die Kopfhörerverstärker auf die Kopfhörer geleitet. Mithilfe eines Kunstkopfes wird der Pegel für jeden Kopfhörer im System gemessen. Durch die Lautstärkeregelung der Kopfhörerverstärker wird für alle Kopfhörer der Pegel auf ein einheitliches Niveau gebracht. Zur Feinabstimmung ist ein weiterer JACK-Mixer in der JACK-Umgebung vorgesehen, welcher zusätzlich noch den Pegel für die rechte und linke Seite eines jeden Kopfhörers einzeln einstellen kann. Somit können auch etwaige Abweichungen zwischen dem rechten und linken Kanal der Wiedergabekette kompensiert werden. Die ganze Prozedur muss für jeden Kopfhörer mit dem entsprechenden Rendering-PC und der angeschlossenen Soundkarte durchgeführt werden, da die unterschiedlichen Soundkarten auch unterschiedliche Ausgangspegel aufweisen können.

Nach dieser grundsätzlichen Vereinheitlichung aller Ausgangspegel kann das Verhältnis zwischen Diffus- und Direktschall eingestellt werden. Da die BRIRs alle ohne Direktschall berechnet wurden, ist der Anhaltspunkt für das Diffus-/Direktschallverhältnis in der Simulation nicht vorhanden. Aus diesem Grund wurde in RAVEN zusätzlich für die Kalibrierung ein Modell des reflexionsarmen Raumes simuliert. Dieses Modell besteht aus einem Raum, in dem alle Oberflächen 100% Absorption aufweisen. In ihm wurden alle vier Quellen in ihrer in Abschnitt 3.1.4 bestimmten Anordnung platziert. Zusätzlich wurde ein binauraler Empfänger in 5 m Entfernung von dieser Anordnung eingefügt. Für diesen wurden BRIRs für alle vier Quellen mit dem Direktschall berechnet. In diesen BRIRs hat also nur die Luftdämpfung Einfluss auf den Direktschall. Im Laboraufbau wurde an der gleichen Position im Raum ein Kunstkopf aufgestellt (s. Abbildung F.5). Seine Empfängerposition im Schallfeld stimmt also mit der virtuellen Position überein. Die Direktschallpegel können verglichen werden, wobei der Direktschall im realen Raum als Referenz gilt und der Direktschall in der virtuellen Umgebung an diesen Pegel angepasst werden kann.

Die Kalibrierung der Direktschallpegel wurde, wie in Abbildung 3.14 dargestellt, mit der Methode von Stahnke (2013) durchgeführt: Zunächst wurde ein Musiker

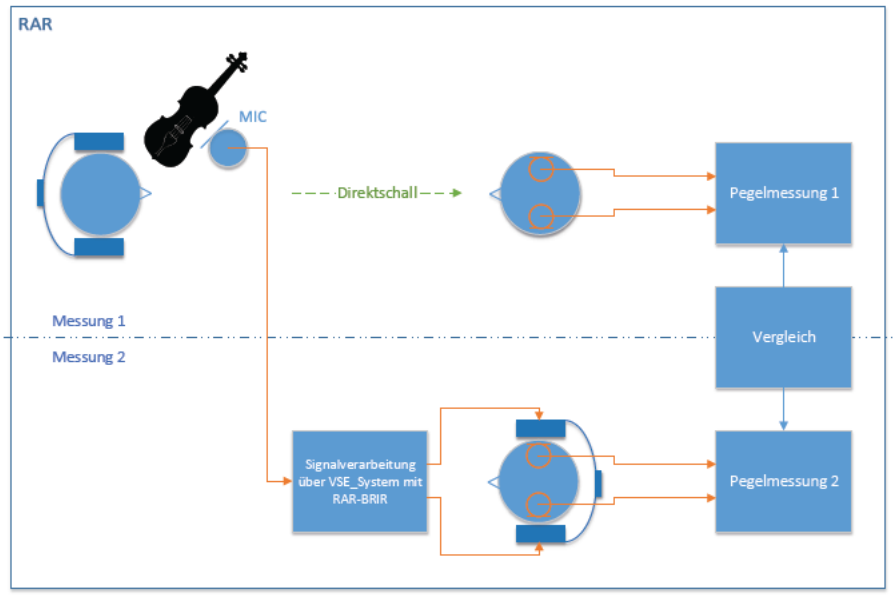


Abbildung 3.14.: Kalibrierung der Pegel der Auralisation im Vergleich zum realen Schallfeld.

aufgefordert einen sehr kurzen Ton und nach einer kleinen Pause einen lang gehaltenen Ton zu spielen. Das Signal wurde mit dem Kunstkopf auf einem Computer mit einer Soundkarte und DAW aufgezeichnet. Daraufhin wurde der RMS-Pegel der Aufnahme des langen Tons bestimmt. Gleichzeitig wurde das Signal auch über das Mikrofon direkt am Instrument auf dem Rendering-PC aufgezeichnet. Diese Aufzeichnung ist ebenfalls über eine Routine in das VSE-Skript implementiert und kann über das Textmenü aufgerufen werden. Im zweiten Schritt wird das auf dem Rendering PC aufgezeichnete Signal durch das Wiedergabesystem über Kopfhörer, welche dem Kunstkopf aufgesetzt wurden, abgespielt. Als BRIR für den Renderer kommt hier die zuvor berechnete BRIR des RAR zum Einsatz. Auch hiervon wird mit dem Kunstkopf der RMS-Pegel bestimmt. Wichtig ist hier, das der Kopfhörer bereits bei der ersten Messung des Direktschalls vom Instrument auf dem Kunstkopf platziert wird. Der Fehler in der Pegelmessung, der durch das Auf- und Absetzen hervorgerufen werden kann, ist größer als der Einfluss des Kopfhörers auf das Schallfeld am Kunstkopf. Nun liegen für die Position des Kunstkopfes Aufnahmen des Direktschalls direkt vom Instrument und für den künstlich durch die VAE erzeugten Direktschall vor. Diese Werte werden verglichen und ihre Differenz bestimmt. Der kurze Ton am Anfang dient der exakten Ausrichtung der beiden Audiosignale zueinander. So kann der Pegel über denselben Bereich beider Audioaufnahmen unabhängig von der Länge der Aufnahme bestimmt werden. Mit der ermittelten Differenz wird der Pegel der künstlichen Wiedergabe eingestellt, um denselben Pegel an der Hörposition zu rekonstruieren, den der Direktschall in der Realität erreicht.

Die Pegeleinstellung geschieht über den JACK-Mixer am Eingang. Das zuvor auf einen Wert mit möglichst idealem Rauschabstand ausgesteuerte Mikrofonsignal

wird direkt nach der AD-Wandlung mit dem ermittelten Wert aus der Kalibrierung gepegelt. Da der Eingangssignalpegel für einen Empfänger im virtuellen Schallfeld korrekt eingestellt ist, ist er durch die vergleichbaren BRIRs für alle Empfänger korrekt. Die Messung muss für alle Quellen wiederholt werden, ist jedoch immer für alle Empfänger gültig. Auf diese Weise wurde das System kalibriert und kann nun unabhängig von der tatsächlich produzierten Lautstärke durch die Instrumente mit dem korrekten Diffus-/Direktschallverhältnis betrieben werden.

Der komplette Ablauf der Kalibrierung ist in Abbildung 3.15 aufgelistet. Alle Funktionen können konsekutiv über das Textmenü des VSE-Skripts ausgeführt werden. Für den Kopfhörer an der Position des Kunstkopfes wurde aus organisatorischen Gründen im Messaufbau ein weiterer ebenfalls korrekt kalibrierter Kopfhörerverstärker eingesetzt. Die technische Signalführung ist in Abbildung 3.9 dargestellt.

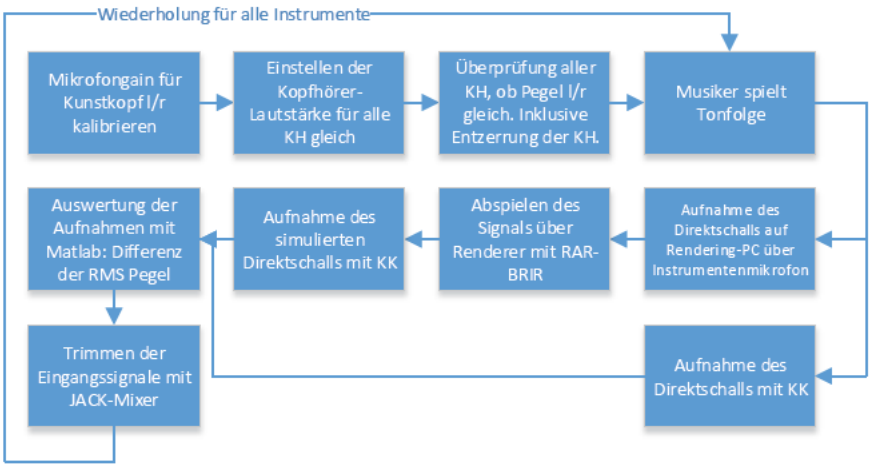


Abbildung 3.15.: Ablauf der Pegelkalibrierung des Wiedergabesystems der VAE

3.2.7. Systemlatenz

Ein weiterer wichtiger Faktor für die Akzeptanz der simulierten akustischen Umgebung durch die Musiker ist die Latenz des Systems. Aus der Aufnahme, Signalverarbeitung und Wiedergabe ergibt sich eine Verzögerung im Audiosignal, welche die Wahrnehmungsgrenze nicht überschreiten darf. Anhaltspunkte für Wahrnehmungsgrenzen von Latenzzeiten bei der Auralisation von virtuellen akustischen Umgebungen gibt Lindau (2009). Er stellte fest, dass die Wahrnehmungsschwelle für die Gesamt-System-Latenz im Mittel bei 107 ms liegt während die kleinste wahrgenommene Latenz bei 53 ms lag.

Für die Messung der Audiolatenz wurde die Impulsantwort des Systems vom Mikrofoneingang der Soundkarte durch alle Signalverarbeitungsschritte bis zum Ausgang

an der Soundkarte gemessen. Bei einem kalibrierten Messsystem ist die gemessene Verzögerung des Signals durch den Prüfling direkt an der Verschiebung der Impulsantwort im Zeitbereich abzulesen. Als Messsystem kam ein Computer mit Matlab und der ITA Toolbox[22], einer von der RWTH Aachen entwickelten Messumgebung für Matlab sowie eine Presonus Firebox als Soundkarte zum Einsatz.

Um die Messung unabhängig vom Einfluss der in der Signalverarbeitung eingesetzten Kompensationsfilter und der Raumimpulsantwort des Renderers durchführen zu können, wurden für die Kopfhörer- und Mikrofonentzerrung einfache Dirac-Impulse ohne Verzögerung als Impulsantwort verwendet. Für die im Renderer verwendete BRIR wurde eine eigene SOFA-Datei erzeugt, die für alle Raumrichtungen ebenfalls Dirac-Impulse enthielt und so keine Verzögerung des Signals bewirkte. Auf diese Weise war es möglich, die Latenz mit allen laufenden Softwarekomponenten und somit Rechenoperationen durchzuführen, welche die Systemlatenz beeinflussen können. Dabei wurde jedoch der Einfluss der später verwendeten Filter eliminiert. Neben allen Signalverarbeitungsschritten wurde auch das JACK-Routing wie im späteren Versuch eingestellt. Die Anzahl an Audioverbindungen innerhalb der JACK-Umgebung hat entscheidenden Einfluss auf die Stabilität der Audioübertragung. Sie hängt direkt mit dem Sample-Buffer der Soundkartentreiber zusammen. Um so höher der Buffer eingestellt ist, desto mehr Verbindungen in der Jack-Umgebung sind möglich, desto größer wird jedoch auch die Latenz. Für die hier benötigte Anzahl an Verbindungen ergab sich eine Puffergröße von 512 Samples als bester Kompromiss zwischen Latenz und Stabilität. Dabei unterschieden sich die beiden verwendeten Soundkarten leicht in ihrer Latenz. Die Messung des Systems unter Verwendung der RME Fireface Karte ist in Abbildung 3.16 dargestellt. Diese resultierende Latenz wurde noch um 176 Samples erweitert, um denselben Wert zu erhalten wie die Latenz des Systems mit der Focusrite Scarlett Soundkarte, die eine 4 ms längere Latenz von 52 ms bei einer Puffergröße von 512 Samples aufwies.

Um die Latenz weiter zu reduzieren wurden die für den Versuch vorgesehenen BRIRs auf ihre Verzögerung untersucht. Dadurch, dass es sich um Raumimpulsantworten handelte, hängt deren Verzögerung des Maximalwerts von der Laufzeit der ersten Reflexionen ab. Dieser Umstand kann ausgenutzt werden, um einen Teil der Systemlatenz zu kompensieren. Die Laufzeit der Reflexionen in der VAE ergibt sich aus der Gesamt-Systemlatenz addiert zur Verzögerung der berechneten Reflexionen. Die BRIRs können also am Anfang um den Wert der Systemlatenz gekürzt werden, ohne die ursprünglich berechnete Laufzeit des Schalls im virtuellen Raum zu beeinflussen. Die Kürzung darf selbstverständlich keine relevanten Informationen der BRIR betreffen. Als Kriterium für die Ermittlung der zu kürzenden Samples wurde ein Wert bestimmt, bei dem die Amplitude der Impulsantwort nahe bei 0 liegt. Für die ermittelten Amplituden ergab sich 10^{-8} als geeignet. Für alle BRIR Datensätze wurde das letzte mögliche Sample bestimmt, was einen Wert kleiner als 10^{-8} aufwies. Die geringste ermittelte Anzahl an zu kürzenden Samples

[22] http://www.ita-toolbox.org/

wurde für die Kürzung aller BRIR-Datensätze verwendet. Hieraus ergab sich eine Verringerung der Systemlatenz um 10 ms.

Der resultierende Wert von 42 ms liegt unter der zuvor beschriebenen minimalen Wahrnehmungsgrenze von 53 ms für die Gesamt-System-Latenz und ist somit für den Versuchsaufbau geeignet. Auch bei geringfügig größeren Verzögerungen zum Beispiel durch die Wahl der BRIR anhand der Tracking-Daten ist dieser Wert weit von dem Mittelwert von 107 ms als JND entfernt.

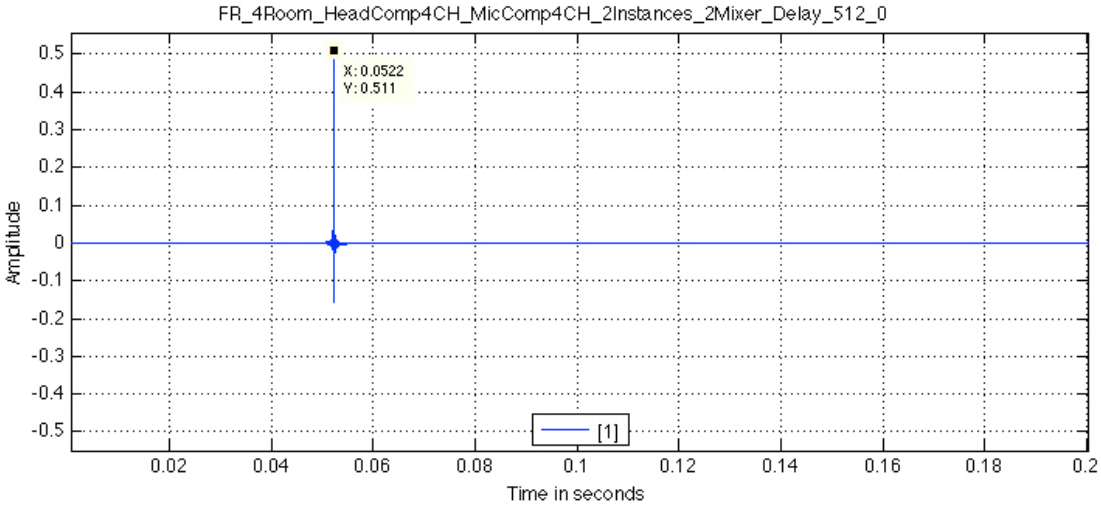


Abbildung 3.16.: Impulsantwort des Rendering-Systems bei Verwendung der RME Fireface Soundkarte. An der Impulsantwort kann die Latenz des gemessenen Systems abgelesen werden.

4. Evaluation der Versuchsumgebung im Vorversuch

Die entwickelte Versuchsumgebung wurde in einem Vorversuch auf ihre Funktionalität und die Akzeptanz durch die teilnehmenden Musiker mit zwei Probanden geprüft. Dabei stand der störungsfreie Betrieb, die Einschätzung der Realitätsnähe des Systems durch die Musiker und der Grad der Immersion im Mittelpunkt der Evaluation. Hierzu wurde ein kurzer Interviewleitfaden entwickelt, mit dessen Hilfe die Probanden befragt wurden.

4.1. Durchführung

Der gesamte Versuchsaufbau wurde im reflexionsarmen Raum installiert. Beide PCs wurden im Verbund mit vier laufenden Software-Renderern und der entsprechenden Hardware aufgebaut und getestet. Im Versuch mit den Musikern wurden jedoch nur zwei Kopfhörer und die entsprechenden physikalischen Ausgänge aktiv verwendet, da nur zwei Musiker für den Vorversuch zur Verfügung standen. Zur Evaluation der Funktionsweise des Systems war dies jedoch ausreichend, da eine tatsächliche Analyse der gemeinsamen Spielweise der Musiker nicht Teil dieser Arbeit war. Der Aufbau des Systems im RAR ist in Abbildung 4.1 abgebildet. In Abbildung F.2 und F.3 im Anhang werden weitere Bilder des Versuchsaufbaus und der Durchführung aufgeführt.

Abbildung 4.1.: Versuchsaufbau im reflexionsarmen Raum mit zwei Plätzen für zwei Musiker und den extra-auralen Kopfhörern. Im Hintergrund der Anordnung befindet sich der Kunstkopf.

Das Versuchsdesign des Vorversuchs sah folgende Schritte vor:

- Kalibrierung der Ausgangspegel
- Einführung der Musiker in den Versuch
- Kalibrierung des Diffus/Direktschallverhältnisses
- 1. Durchlauf mit Auralisation von vier Räumen
- 1. Interview
- 2. Durchlauf mit Auralisation von vier Räumen
- 2. Interview
- 3. Durchlauf mit Auralisation von vier Räumen
- 3. Interview
- 4. Durchlauf mit Auralisation von drei Räumen
- 4. Interview
- Befragung zu Alter und Konzerterfahrung.
- Unterzeichnung der Einverständniserklärung zur Verwendung der Daten.

Die Kalibrierung der Ausgangspegel des Systems wurde in der Vorbereitung analog zum in Abschnitt 3.2.6 beschriebenen Verfahren durchgeführt. Hier war die Anwesenheit der Musiker noch nicht notwendig. Das Diffus-/Direktschallverhältnis wurde dann als erstes zusammen mit den Musikern und ihren Instrumenten eingestellt.

Jeder Durchlauf der Auralisation wurde mit vier Raummodellen durchgeführt. Die Musiker wurden gebeten, einen kurzen Ausschnitt eines Stücks von ein bis zwei Minuten Dauer zu spielen. Dabei sollten sie sich auf ihr gemeinsames Spiel konzentrieren und versuchen den Einfluss der Akustik und des technischen Aufbaus auf ihr Zusammenspiel zu beobachten. Sobald sie ihr Spiel beendet hatten, wurde der Raum in der Software gewechselt. Nach vier Räumen wurden sie dann anhand des Interviewleitfadens befragt:

- Wie war Ihr erster Eindruck?
- War es schwierig oder ungewohnt in der Umgebung zu spielen? (Akustische und visuelle Reize)
- Haben Sie besondere Unterschiede zwischen den einzelnen Räumen bemerkt?
- Gab es Fehler/Artefakte/Störgeräusche?
- Hat Sie der Versuchsaufbau gestört? Zum Beispiel die Verkabelung, Mikrofone etc.?

Für den Vorversuch konnten zwei Musikerinnen gewonnen werden. Beide sind Studentinnen im Alter von 22 und 29 Jahren. Mehr als 10 Jahre Konzerterfahrung qualifizierten sie zur Teilnahme am Versuch. Eine Musikerin war Bratschistin, eine Klarinettistin. Die Klarinette ist im ursprünglichen Versuchsdesign als Instrument nicht vorgesehen. Zur Überprüfung der Versuchsumgebung auf Fehlerfreiheit und zur Evaluation der Plausibilität der Auralisation war sie jedoch geeignet.

4.2. Ergebnis des Vorversuchs

Das System konnte intensiv und umfangreich getestet werden. Im Verlauf der Versuchsdurchführung stellte sich die Kalibrierung als aufwändigster Teil der Einrichtung des Systems heraus und wurde noch während des Aufbaus und der Durchführung mit den Musikern optimiert.

Die Kalibrierung des Diffus-/Direktschallverhältnisses ergab zunächst falsche Ergebnisse. Während der Kalibrierung wurde zunächst nur das Signal von einem Mikrofon verwendet. Die Verwendung von Kugelmikrofonen und die zwar realitätsnahe, jedoch für die Versuchsumgebung relativ enge Positionierung der Musiker führte dazu, dass der im eigentlichen Versuch aufgenommene Direktschallpegel der Instrumente im Vergleich zum für die Kalibrierung verwendeten Pegel zu hoch war. Hierfür war die Aufnahme des Direktschalls über mehrere, durch Übersprechen beeinflusste Mikrofone während des Versuchs verantwortlich. Es stellte sich heraus, dass auch während der Kalibrierung alle Mikrofone für alle Instrumente bereits an ihrer späteren Position im Raum und am Instrument vorhanden und eingeschaltet sein müssen, um den durch das Übersprechen erhöhten Signalpegel mit in die Kalibrierung aufzunehmen. Hierfür wurden alle Mikrofonsignale für den in Abbildung 3.14 abgebildeten Aufbau und den Ablauf der Kalibrierung in der JACK-Routing-Umgebung zusammengemischt. Dies wurde noch während des Vorversuchs umgesetzt und somit verbessert. Für spätere Versuche mit dem vorgestellten Versuchsaufbau sollten Mikrofone mit Nierencharakteristik verwendet und die Anordnung der Musiker etwas ausgedehnt werden, um das Übersprechen weiter zu reduzieren.

Des Weiteren wurde festgestellt, dass ein einfacher Ton für eine korrekte Erfassung des Direktschallpegels nicht ausreicht, da nicht alle möglichen Frequenzbereiche des Instruments angeregt werden. Durch Unterschiede zwischen der realen und simulierten Richtcharakteristik des Instruments im Frequenzbereich des einzelnen Tons ergibt sich ein Fehler in der Kalibrierung, der durch ein breitbandiges Signal minimiert werden kann. Deshalb wurden die Musiker nach einem ersten Durchgang gebeten, für die Kalibrierung einen kurzen Ton zur Ausrichtung des Signals und dann eine längere Tonfolge über alle Lagen zu spielen. Auf diese Weise wurde der gesamte mögliche Frequenzbereich des Instruments abgedeckt. Durch die zeitliche Mittelung über die gesamte Tonfolge in der Pegelmessung wird der Fehler durch die Abweichungen der Richtcharakteristiken reduziert.

Die eigentliche Auralisation konnte nach erfolgreicher Kalibrierung des Systems wie geplant durchgeführt werden. Kleinere Mängel im technischen Aufbau, wie die Position und Verkabelung der Mikrofone konnten durch Hinweise der Probanden sofort behoben werden. Im Interview wurde schnell deutlich, dass ein hoher Grad an Immersion in die virtuelle akustische Umgebung vorlag und die Musiker sich sofort auf die Eigenschaften der Räume konzentrieren konnten. Es wurden keine Artefakte oder Störgeräusche festgestellt. Die Eindrücke bezogen sich direkt auf den Klang des

eigenen Instruments im Raum und den verursachten Nachhall. Dabei wurden durch die Musiker bereits im ersten Durchgang klare Unterschiede zwischen den Räumen festgestellt. Diese bezogen sich einerseits auf die Nachhallzeit des simulierten Raums, die Schärfe und Färbung des Nachhalls und andererseits auch direkt auf das gemeinsame Spiel und die Probleme oder Vorteile, die aus der unterschiedlichen akustischen Situation entstanden. Dabei konnten die Probanden ohne Aufforderung direkt den Bezug zwischen ihrer Interpretation und den Eigenschaften des Raums herstellen. Bei der Schilderung ihrer Wahrnehmung wurden Begriffe oder Situationen aus ihrer eigenen Konzerterfahrung als Beispiele oder Referenz verwendet. Auf Basis dieser Grundlagen wurden schnell eigene Favoriten in den präsentierten Räumen gefunden, welche das eigene Spiel besonders „schön“ und das Zusammenspiel einfach gestalteten.

5. Zusammenfassung und Diskussion

Mit der vorgestellten Methode zur Berechnung von BRIRs mithilfe von RAVEN und einem dynamischen Matlab Skript konnten geeignete Daten erzeugt werden, um virtuelle akustische Umgebungen zu realisieren. Alle entscheidenden Einflüsse auf das virtuelle Schallfeld konnten dabei beachtet werden. Es wurden Richtcharakteristiken für Quellen und Empfänger implementiert sowie Freiheitsgrade für die Bewegungen der Musiker ermittelt und in die Berechnung integriert. Mit den gewählten Parametern für die Berechnung in der Simulationssoftware wurden geeignete Impulsantworten für eine realitätsnahe Auralisation gewonnen. In einem Vorversuch konnte dabei festgestellt werden, dass die gewonnenen BRIRs als plausibel angesehen werden und keine Artefakte durch die Berechnungsmethode aufweisen.

Wichtig für den Einsatz der Versuchsumgebung ist die einfache Rekonstruktion von früheren Ergebnissen und die schnelle Durchführung von neuen Berechnungen. Diesen Vorgaben wurde, durch die Automatisierung der Berechnungen mit einem Matlab-Skript und die klare Datenstrukturierung mit zusätzlichen Metadaten und Protokolldateien, Rechnung getragen. Auf diese Weise ist bei allen berechneten Daten der Ursprung und die verwendete Methode mit ihren Parametern nachvollziehbar. Die Fehlerkorrektur wird hierdurch erleichtert. Mit der automatisierten Simulation der virtuellen Raummodelle und der damit einhergehenden Berechnung der binauralen Impulsantworten können weitere Datensätze erstellt werden.

Zur Auralisation der virtuellen akustischen Umgebungen mithilfe der zuvor generierten BRIRs wurde ein technischer Aufbau entwickelt. Zwei Computer wurden für die Faltung der BRIRs mit den Quellsignalen der Instrumente verwendet. Mit ihnen wurde die Auralisation der virtuellen Umgebungen in Echtzeit ermöglicht. Die Latenz des Gesamtsystems wurde auf ein Minimum reduziert und konnte auf einen Wert unterhalb der Wahrnehmungsgrenzen begrenzt werden. Technische Einflussfaktoren, wie die Übertragungseigenschaften der Mikrofone und extra-auralen Kopfhörer wurden beachtet und mithilfe digitaler Signalverarbeitung kompensiert. Durch ein Verfahren zur Kalibrierung der Signalpegel in der Auralisation konnte ein realistisches Verhältnis zwischen realem Direktschall und simulierter Raumantwort erzielt werden.

Der technische Aufbau der Versuchsumgebung wurde mit zwei Musikern auf seine Funktionalität und Plausibilität der Auralisation untersucht. Es konnte gezeigt werden, dass das System dazu geeignet ist, eine realistisch wirkende virtuelle Konzertumgebung für Musiker zu erzeugen. Die Musiker waren in der Lage, sich in die Situation hinein zu fühlen und sich auf die wahrgenommene Akustik zu konzentrieren. Dem dafür benötigten Versuchsdesign kommt die Automatisierung aller

Steuerbefehle für die Softwareumgebungen und die Optimierung der Auralisation in Bezug auf die Leistungsfähigkeit des Renderings zugute. Es kann schnell zwischen bis zu vier Räumen gewechselt werden, um direkte Vergleiche zwischen verschiedenen akustischen Umgebungen zu ermöglichen. Der Versuchsleiter wird hierfür durch die Menüführung in den verwendeten Skripten im Versuchsablauf unterstützt.

Mit dem vorgestellten Versuchsaufbau können nun Versuche zur musikalischen Interpretation von Musikern im Zusammenspiel bei unterschiedlicher Raumakustik durchgeführt werden. Das vorgestellte System ist dabei beliebig skalierbar. Eine Erweiterung des Systems um weitere Quellen und Empfänger mit entsprechender zusätzlicher Hardware ist ohne weiteres möglich. Die Nutzung der Versuchsumgebung kann außerdem auf weitere Studienbereiche der Akustik ausgedehnt werden. Nachfolgende Arbeiten am Fachgebiet Audiokommunikation werden sich zum Beispiel mit der Untersuchung der Wahrnehmung und Beschreibung raumakustischer Umgebungen durch Musiker beschäftigen.

Literaturverzeichnis

Ackermann, David und Maximilian Ilse (2015): *The Simulation of Monaural and Binaural Transfer Functions for a Ground Truth for Room acoustical Analysis and Perception (GRAP).* Masterarbeit, Technische Universität, Fakultät I, Fachgebiet Audiokommunikation, Berlin.

Ahnert, Wolfgang; Stefan Feistel und Hans-Peter Tennhardt (2008): „Computer Aided Sound System Design." In: Glen Ballou (Hrg.) *Handbook for Sound Engineers (Fourth Edition).* Oxford: Focal Press, S. 1337 1384.

Ahnert, Wolfgang und Hans-Peter Tennhardt (2008): „Raumakustik." In: Stefan Weinzierl (Hrg.) *Handbuch der Audiotechnik.* Berlin, Heidelberg: Springer-Verlag, S. 181 266.

Algazi, V Ralph; Richard O Duda; Ramani Duraiswami; Nail A Gumerov und Zhihui Tang (2002): „Approximating the head-related transfer function using simple geometric models of the head and torso." In: *J. Acoust. Soc. Am.*, 112(5) S. 2053 2064.

Aspöck, Lukas; Sönke Pelzer; Frank Wefers und Michael Vorländer (2014): „A Real-Time Auralization Plugin for Architectural Design and Education." In: *Proc. of the EAA Joint Symposium on Auralization and Ambisonics.* Berlin, S. 156 161.

Blauert, Jens und Jonas Braasch (2008): „Räumliches Hören." In: Stefan Weinzierl (Hrg.) *Handbuch der Audiotechnik.* Berlin, Heidelberg: Springer-Verlag, S. 87 121.

Bolzinger, S; O Warusfel und E Kahle (1994): „A study of the influence of room acoustics on piano performance." In: *J. Phys. IV*, Colloque C5, supplément au Journal de Physique III 4 S. 617 620.

Brinkmann, Fabian; Alexander Lindau; Stefan Weinzierl; Gunnar Geissler und Steven van de Par (2013): „A high resolution head-related transfer function database including different orientations of head above the torso." In: *Fortschritte der Akustik: Tagungsband der 39. DAGA.* Merano, Italy, S. 596 599.

Burdea, Grigore und Philippe Coiffet (1994): „Introduction." In: *Virtual reality technology.* New York u.a.: Wiley, S. 1 15.

Chéenne, Dominique J. (2008): „Modeling and Auralization." In: Glen Ballou (Hrg.) *Handbook for Sound Engineers (Fourth Edition).* Oxford: Focal Press, S. 213 237.

Geier, Matthias; Jens Ahrens und Sascha Spors (2008a): „ASDF: Ein XML Format zur Beschreibung von virtuellen 3D Audioszenen." In: *Fortschritte der Akustik: Tagungsband der 34. DAGA.* Dresden.

Geier, Matthias; Jens Ahrens und Sascha Spors (2008b): „The SoundScape Renderer: A Unified Spatial Audio Reproduction Framework for Arbitrary Rendering Methods." In: *Audio Engineering Society Convention 124*, 7330. Amsterdam, The Netherlands.

Geier, Matthias und Sascha Spors (2012): „Spatial Audio with the SoundScape Renderer." In: *27. Tonmeistertagung - VDT International Convention*. Köln.

Goldstein, Robert A und Roger Nagel (1971): „3-D Visual simulation." In: *Simulation*, 16(1) S. 25 31.

Kato, Kosuke; Kanako Ueno und Keiji Kawai (2007): „Musicians Adjustment of Performance to Room Acoustics Part II: Acoustical Analysis of Performed Sound Signals." In: *Proceedings of the 19th International Congress on Acoustics*. Madrid, Spain.

Kato, Kosuke; Kanako Ueno und Keiji Kawai (2008): „Musicians' Adjustment of Performance to Room Acoustics, Part III: Understanding the Variations in Musical Expressions." In: *Proc. of Acoustics 08 Paris*. S. 4533 4538.

Kuttruff, Heinrich (2009): *Room Acoustics*. 5. Auflage. New York: Taylor & Francis.

Lerch, Reinhard; Gerhard Martin Sessler und Dietrich Wolf (2009): „Numerische Verfahren der Akustik - Computational Acoustics." In: *Technische Akustik*. Berlin, Heidelberg: Springer-Verlag, S. 758 820.

Lindau, Alexander (2009): „The Perception of System Latency in Dynamic Binaural Synthesis." In: *Fortschritte der Akustik: Tagungsband der 35. DAGA*. Rotterdam, S. 1063 1066.

Lindau, Alexander (2014): *Binaural Resynthesis of Acoustical Environments. Technology and Perceptual Evaluation*. Dissertation, Technische Universität, Fakultät I - Geisteswissenschaften, Berlin.

Lindau, Alexander und Fabian Brinkmann (2012): „Perceptual evaluation of headphone compensation in binaural synthesis based on non-individual recordings." In: *J. Audio Eng. Soc*, 60(1/2) S. 54 62.

Lindau, Alexander; Torben Hohn und Stefan Weinzierl (2007): „Binaural resynthesis for comparative studies of acoustical environments." In: *Audio Engineering Society Convention 122*, 7032. Vienna, Austria.

Lindau, Alexander; Linda Kosanke und Stefan Weinzierl (2012): „Perceptual evaluation of model-and signal-based predictors of the mixing time in binaural room impulse responses." In: *J. Audio Eng. Soc*, 60(11) S. 887 898.

Lindau, Alexander; Hans-Joachim Mempel und Stefan Weinzierl (2008): „Minimum BRIR grid resolution for dynamic binaural synthesis." In: *Proc. of the Acoustics 08 Paris*. Paris, S. 3851 3856.

Lindau, Alexander und Stefan Weinzierl (2007): „FABIAN - Schnelle Erfassung binauraler Raumimpulsantworten in mehreren Freiheitsgraden." In: *Fortschritte der Akustik: Tagungsband der 33. DAGA*. Stuttgart.

Lindau, Alexander und Stefan Weinzierl (2012): „Assessing the Plausibility of Virtual Acoustic Environments." In: *Acta Acustica united with Acustica*, 98(5) S. 804 810.

Majdak, Piotr et al. (2013): „Spatially oriented format for acoustics: A data exchange format representing head-related transfer functions." In: *Audio Engineering Society Convention 134*, 8880. Rome, Italy.

Moldrzyk, Christoph; Tobias Lentz und Stefan Weinzierl (2005): „Perzeptive evaluation binauraler Auralisationen." In: *Fortschritte der Akustik: Tagungsband der 31. DAGA*. München, S. 545 546.

Møller, Henrik (1992): „Fundamentals of Binaural Technology." In: *Applied Acoustics*, 36(3) S. 171 218.

Møller, Henrik; Dorte Hammershøi; Clemen Boje Jensen und Michael Friis Sørensen (1995): „Transfer Characteristics of Headphones Measured on Human Ears." In: *J. Audio Eng. Soc*, 43(4) S. 203 217.

Möser, Michael (2012): „Eigenschaften und Beschreibung von Übertragern." In: *Technische Akustik*, Kap. 13. Springer-Verlag Berlin Heidelberg.

Pollow, Martin; Gottfried K Behler und Frank Schultz (2010): „Musical instrument recording for building a directivity database." In: *Fortschritte der Akustik: Tagungsband der 36. DAGA*.

Popko, Reinhild (2013): *Zur Hörbarkeit und Interpolation von Kopf-über-Torso-Orientierungen bei Aufnahmen mit einem Kopf-und-Torso-Simulator*. Bachelorarbeit, Technische Universität, Fakultät I, Fachgebiet Audiokommunikation, Berlin.

Rindel, Jens Holger (1995): „Computer simulation techniques for acoustical design of rooms." In: *Acoustics Australia*, 23 S. 81 86.

Schärer Kalkandjiev, Zora und Stefan Weinzierl (2013): „The influence of room acoustics on solo music performance: An empirical case study." In: *Acta Acustica united with Acustica*, 99(3) S. 433 441.

Schärer Kalkandjiev, Zora und Stefan Weinzierl (2015): „The Influence of Room Acoustics on Solo Music Performance: An Experimental Study." In: *Psychomusicology: Music, Mind, and Brain*, Advance online publication. http://dx.doi.org/10.1037/pmu0000065.

Schöps, P; N Seichert; M Schenk; U Petri und E Senn (1997): „Alters- und geschlechtsspezifische Bewegungsausmaße der Halswirbelsäule." In: *Physikalische Medizin, Rehabilitationsmedizin, Kurortmedizin*, 7(03) S. 80 87.

Schröder, Dirk (2011): *Physically Based Real-Time Auralization of Interactive Virtual Environments*. Dissertation, Rheinisch-Westfälische Technische Hochschule, Fakultät für Elektrotechnik und Informationstechnik, Aachen.

Schröder, Dirk; Philipp Dross und Michael Vorländer (2007): „A fast reverberation estimator for virtual environments." In: *Audio Engineering Society Conference: 30th International Conference: Intelligent Audio Environments*. Saariselkä, Finland.

Schröder, Dirk und Michael Vorländer (2011): „RAVEN: A Real-Time Framework for the Auralization of Interactive Virtual Environments." In: *Forum Acusticum 2011*. Aalborg, Denmark, S. 1541 1546.

Slavik, Karl M und Stefan Weinzierl (2008): „Wiedergabeverfahren." In: Stefan Weinzierl (Hrg.) *Handbuch der Audiotechnik*. Berlin, Heidelberg: Springer-Verlag, S. 609 685.

Stahnke, Erik (2013): *Über die raumakustische Wahrnehmung durch Musiker*. Masterarbeit, Technische Universität, Fakultät I, Fachgebiet Audiokommunikation, Berlin.

Thurlow, Willard R.; John W. Mangels und Philip S. Runge (1967): „Head Movements During Sound Localization." In: *J. Acoust. Soc. Am.*, 42(2) S. 489 493.

Ueno, Kanako; Kosuke Kato und Keiji Kawai (2010): „Effect of Room Acoustics on Musicians' Performance. Part I: Experimental Investigation with a Conceptual Model." In: *Acta Acustica united with Acustica*, 96 S. 505 515.

Von Békésy, Georg (1968): „Feedback phenomena between the stringed instrument and the musician." In: *The Rockefeller University Review*, 6.

Vorländer, Michael (2007): „Introduction." In: *Auralization: Fundamentals of Acoustics, Modelling, Simulation, Algorithms and Acoustic Virtual Reality*. Berlin, Heidelberg: Springer-Verlag, S. 1 5.

Weinzierl, Stefan (2008): „Aufnahmeverfahren." In: Stefan Weinzierl (Hrg.) *Handbuch der Audiotechnik*, Kap. 10. Berlin, Heidelberg: Springer-Verlag, S. 551 607.

Abbildungsverzeichnis

2.1. Übertragungspfad des Außenohrs (Weinzierl, 2008, S. 586). 11
2.2. Datenbasierte Binauralsynthese 13
2.3. Spiegelschallquellenmodell (Chéenne, 2008, S. 221). 15
2.4. Konstruktion der Impulsantwort mit der hybriden Methode 20

3.1. Koordinatensysteme von SketchUp und RAVEN 25
3.2. Streuung der mittleren Nachhallzeiten 27
3.3. Im Versuch verwendete Modelle (1-8). 28
3.4. Im Versuch verwendete Modelle (9-12). 29
3.5. Positionen der Musiker als Quellen und Empfänger im CAD-Modell 31
3.6. Darstellung der Richtcharakteristik einer Violine als Balloonplot. . . 32
3.7. Programmablaufplan des VSE Matlab-Skripts 35
3.8. Aufnahme-/Wiedergabekette des Versuchsaufbaus im RAR. 47
3.9. Technischer Versuchsaufbau mit allen Signalverbindungen. 48
3.10. Signalverarbeitung des Rendering-Computers im VSE-System. . . . 49
3.11. Mikrofonierung und Kopfhörer im Vorversuch mit einer Bratsche. . 50
3.12. Flussdiagramm Signalverarbeitung 52
3.13. Bildschirmfoto der VSE Softwareumgebung 55
3.14. Pegelkalibrierung 58
3.15. Ablauf der Pegelkalibrierung 59
3.16. Impulsantwort des Rendering-Systems 61

4.1. Versuchsaufbau im reflexionsarmen Raum 63

C.1. Ordnerstruktur der Raven Projektdateien: MATLAB 83
C.2. Ordnerstruktur der Raven Projektdateien: INPUT 84
C.3. Ordnerstruktur der Raven Projektdateien: OUTPUT 85
C.4. Ordnerstruktur der VSE Versuchsumgebung 86

D.1. Bildschirmfoto: Auslastung eines Computers 87
D.2. Reflektogramm für eine Sender/Empfänger-Kombination 88
D.3. Visualisierung der zum Modell absoluten Quell- und Empfängerpositionen 88

F.1. Kalibrierung der Miniaturkondensatormikrofone. 91
F.2. Kontrollraum 92
F.3. Musiker während des Versuchs im RAR. 92
F.4. Extra-aurale Kopfhörer AKG K1000 mit angebautem Headtracker. . 92
F.5. Kunstkopf für die Pegelkalibrierung 93

G.1. Inhalt des beiliegenden Datenträgers 95

Tabellenverzeichnis

3.1. Auflistung und Parameter der verwendeten Räume 27
3.2. Koordinaten der verwendeten Instrumente 30
3.3. Hörtypischer und maximaler Bewegungsbereich (Popko, 2013). . . . 33
3.4. Liste der vom Skript unterstützen Export-Varianten 39

E.1. Patchplan RME Fireface UFX 89
E.2. Patchplan Focusrite Scarlett 18i20 90

A. Ausschnitte Matlab Code

```
%% GLOBAL SKRIPT SETTINGS -----------------------------------------------------

% Calc new BRIR Data (No=0, Yes=1)
calcBRIRData = 1;
% Save MAT file containing BRIR data to disk (No=0, Yes=1)
saveMAT = 0;
% Write BRIR Data to SOFA files (No=0, Yes=1)
exportSOFA = 0;
% Write BRIR Data to single wav files (No=0, Yes=1)
exportWAV = 0;
% Create Wonder Files (No=0, Yes=1)
exportWonder = 0;
% Create SSR File (old Version, only Azimuth), (No=0, Yes=1)
exportSSR720 = 0;
% Calc example convolution (No=0, Yes=1)
calcConvolution = 0;
% Sound file for Convolution
audioConvolution = 'C:\ITASoftware\Raven\RavenDatabase\SoundDatabase\around_midnight.wav';
% Plot Reflectogram for every possible S/R combination (No=0, Yes=1)
plotReflectogram = 0;
% Plot Positons in 3D (No=0, Yes=1)
plotPositions = 0;
% Plot Spectrum (No=0, Yes=1)
plotSpectrum = 0;
```

Quellcode A.1: Globale Einstellungen für das Matlab VSE Skript.

```
function VSE_writeAnimPath( fileID, rp, Fs, blockSize, diracInterval, aziIndex, eleIndex )
%VCH_WRITEANIMPATH Writes an Raven animation path to File
%
%   Input arguments:
%   fileID          = File ID for animation path file
%   rp              = Receiver position, view-vector and up-vector
%   Fs              = Sample rate
%   blockSize       = Block size
%   diracInterval   = Dirac interval
%   aziIndex        = Azimuth angles as continous vector
%   eleIndex        = Elevation angles as continous vector
%
%   -----------------------------------------
%   Christoph Boehm 2015
%   E-Mail: christoph.boehm@campus.tu-berlin.de
%   Revision: 02.03.2015
%

    %% SETTINGS

    debug = 0;      % Activate debug mode (1 = ON)
                    % In debug mode every line of the animation path file
                    % contains the frame index, azimuth and elevation angle
                    % in degrees. Degrees are absolute, not relativ to the
                    % initial view vector.


```

```
28      %% CALCULATION
29
30      %- Debug Output: Total Steps
31      if debug == 1; fprintf('Total Steps: %d\n', length(aziIndex) *
            length(eleIndex)); end      % Print for debug
32
33      % Init indeces
34      frameIndex = 0; % Init frame index
35      init = 0;       % Set init for first line with time = 0
36
37      %- Transform Cartesian coordinates to sphere
38      [Azi, Ele, r] = cart2sph(rp(1,4), rp(1,6), rp(1,5));
39
40      %- Print first line to file (selects Raven or Sketchup Coordinates)
41      fprintf(fileID,'%s\n','Raven');
42
43      % Loop
44      for indexA = 1:length(aziIndex)
45
46          % Set azimuth angle to first angle of aziIndex
47          azi = Azi + aziIndex(indexA)*pi/180;
48
49          for indexE = 1:length(eleIndex)
50
51              % Set elevation angle to first angle of eleIndex
52              ele = Ele + eleIndex(indexE)*pi/180;
53
54              % Print current angle if debug is active
55              if debug == 1; disp(azi*180/pi); disp(ele*180/pi); end
56
57              % Transform polar coordinates back to Cartesian
58              [rp(1,4), rp(1,6), rp(1,5)] = sph2cart(azi, ele, r);
59
60              % Print to file
61              if
62                 debug == 1; fprintf(fileID,'%f, azi: %f, ele: %f, ',frameIndex,
                       azi*180/pi, ele*180/pi);
63              end
64              if
65                 debug == 1; fprintf(fileID,'%d,
                       ',init*(frameIndex*length(diracInterval)
                       +blockSize+blockSize/2));
66              end
67              fprintf(fileID,'%6.12f, ',init*(frameIndex*length(diracInterval)
                    +blockSize+blockSize/2)/ Fs);
68              fprintf(fileID,'%f, ',rp(1,1:end-1));
69              fprintf(fileID,'%f\n',rp(1,end));
70
71              if
72                 debug == 1; fprintf(fileID,'%f, azi: %f, ele: %f, ',frameIndex,
                       azi*180/pi, ele*180/pi);
73              end
74              if
75                 debug == 1; fprintf(fileID,'%d,
                       ',((frameIndex+1)*length(diracInterval)
                       +blockSize+blockSize/2-1));
76              end
77              fprintf(fileID,'%6.12f, ',((frameIndex+1)*length(diracInterval)
                    +blockSize+blockSize/2-1)/Fs);
78              fprintf(fileID,'%f, ',rp(1,1:end-1));
79              fprintf(fileID,'%f\n',rp(1,end));
80
81              % Increment frameIndex
82              frameIndex = frameIndex + 1;
83
84              % After first run init remains 1
```

```
85                 init = 1;
86
87             end
88
89         end
90     end
```

Quellcode A.2: Die Funktion *VSE_writeAnimPath* generiert den Animationspfad für das RAVEN Animationsmodul.

B. VSE Konfigurationsdateien

```
#  ------------------------------------------------------------------------------
#
#
# Jconv headphone configuration
# ------------------------------------------------------------------------------
#
#                 in  out   partition    maxsize      density
# ------------------------------------------------------------------------------
/convolver/new     4    4         512      20480         1.0
#
#
#                in out  gain delay  offset  length  chan  file
# ------------------------------------------------------------------------------
/impulse/read    1   1   1    176       0       0     1    filter_K1000_TLabs.wav
/impulse/read    2   2   1    176       0       0     2    filter_K1000_TLabs.wav
/impulse/read    3   3   1    176       0       0     1    filter_K1000_2377.wav
/impulse/read    4   4   1    176       0       0     2    filter_K1000_2377.wav
```

Quellcode B.1: Konfigurationsdatei für das Programm jConvolver. Es legt die Impulsantworten zur Entzerrung der Kopfhörerfrequenzgänge fest. Da jeder PC nur zwei Kopfhörer verwaltet sind hier auch nur zwei Kopfhörer mit jeweils linkem und rechtem Kanal in einer WAV-Datei angegeben. Das Delay von 176 Samples wird nur für die RME Soundkarte verwendet und kompensiert so die längere Latenz der Focusrite Soundkarte.

```
#  ------------------------------------------------------------------------------
#
#
# Jconv microphone configuration
# ------------------------------------------------------------------------------
#
#                 in  out   partition    maxsize      density
# ------------------------------------------------------------------------------
/convolver/new     4    4         512      20480         1.0
#
#
#                in out  gain delay  offset  length  chan  file
# ------------------------------------------------------------------------------
/impulse/read    1   1   1      0       0       0     1    filter_MKE1_2381.wav
/impulse/read    2   2   1      0       0       0     1    filter_MKE1_2382.wav
/impulse/read    3   3   1      0       0       0     1    filter_MKE1_2383.wav
/impulse/read    4   4   1      0       0       0     1    filter_MKE1_2384.wav
```

Quellcode B.2: Konfigurationsdatei für das Programm jConvolver. Es legt die Impulsantworten zur Entzerrung der Mikrofonfrequenzgänge fest.

```
<?xml version="1.0" ?>
<jack_mixer geometry="297x309">
   <input_channel balance="0.000000" balance_midi_cc="12" name="in_1" type="mono"
       volume="0.000000" volume_midi_cc="11"/>
   <input_channel balance="0.000000" balance_midi_cc="14" name="in_2" type="mono"
       volume="0.000000" volume_midi_cc="13"/>
   <input_channel balance="0.000000" balance_midi_cc="16" name="in_3" type="mono"
       volume="0.000000" volume_midi_cc="15"/>
   <input_channel balance="0.000000" balance_midi_cc="18" name="in_4" type="mono"
       volume="0.000000" volume_midi_cc="17"/>
   <main_mix_channel balance="0.000000" balance_midi_cc="8" volume="0.000000"
       volume_midi_cc="7"/>
</jack_mixer>
```

Quellcode B.3: Konfigurationsdatei für den Jack-Mixer zur Kalibrierung der Eingangssignale.

```
<?xml version="1.0" ?>
<jack_mixer geometry="297x309">
   <input_channel balance="0.000000" balance_midi_cc="20" name="HP1L" type="mono"
       volume="0.000000" volume_midi_cc="19"/>
   <input_channel balance="0.000000" balance_midi_cc="22" name="HP1R" type="mono"
       volume="0.000000" volume_midi_cc="21"/>
   <input_channel balance="0.000000" balance_midi_cc="24" name="HP2L" type="mono"
       volume="0.000000" volume_midi_cc="23"/>
   <input_channel balance="0.000000" balance_midi_cc="26" name="HP2R" type="mono"
       volume="0.000000" volume_midi_cc="25"/>
   <main_mix_channel balance="0.000000" balance_midi_cc="28" volume="0.000000"
       volume_midi_cc="27"/>
</jack_mixer>
```

Quellcode B.4: Konfigurationsdatei für den Jack-Mixer zur Kalibrierung der Ausgangssignale.

```
<jmess>
   <connection>
      <output>system:capture_11</output>
      <input>microphones_eq:In-1</input>
   </connection>
   <connection>
      <output>system:capture_12</output>
      <input>microphones_eq:In-2</input>
   </connection>
   <connection>
      <output>system:capture_13</output>
      <input>microphones_eq:In-3</input>
   </connection>
   <connection>
      <output>system:capture_14</output>
      <input>microphones_eq:In-4</input>
   </connection>
</jmess>
```

Quellcode B.5: Beispielhafte Konfigurationsdatei für das JACK-Routing. Hier werden vier Eingangskanäle der Soundkarte auf die passenden Eingänge des jConvolvers zur Entzerrung der Mikrofonfrequenzgänge geleitet.

C. VSE Ordnerstrukturen

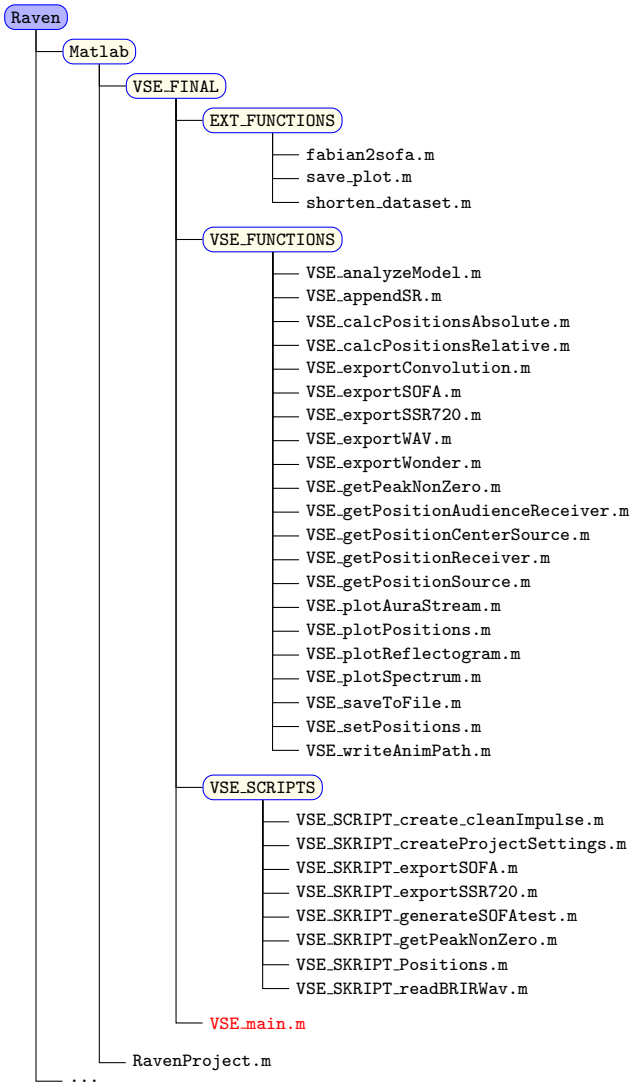


Abbildung C.1.: Ordnerstruktur der Raven Projektdateien: MATLAB. Der Platzhalter „*“ kann beliebige Namenserweiterungen enthalten.

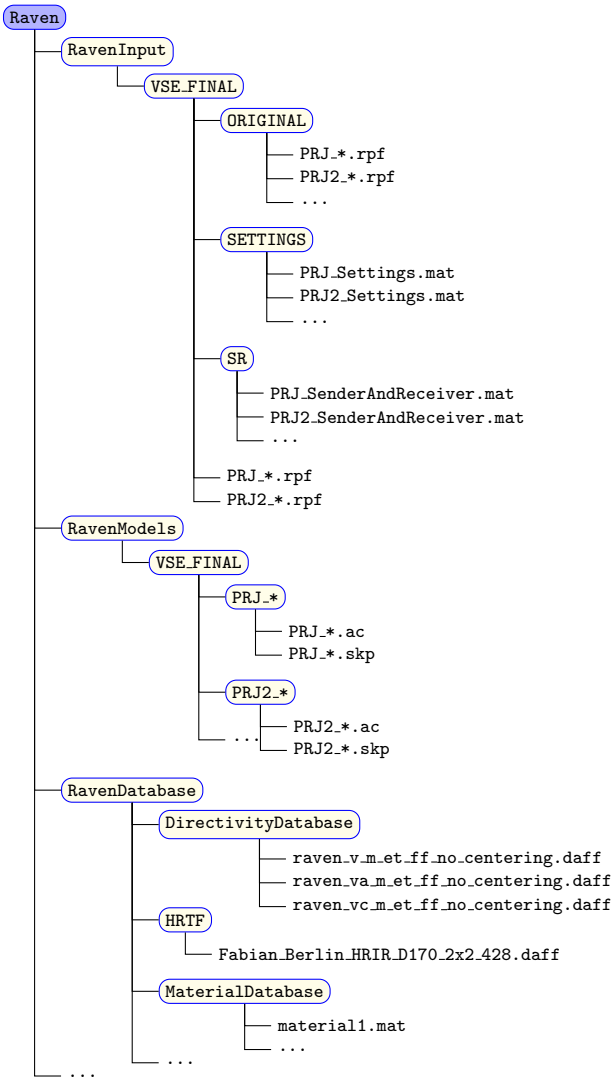


Abbildung C.2.: Ordnerstruktur der Raven Projektdateien: INPUT. Der Platzhalter „*“ kann beliebige Namenserweiterungen enthalten.

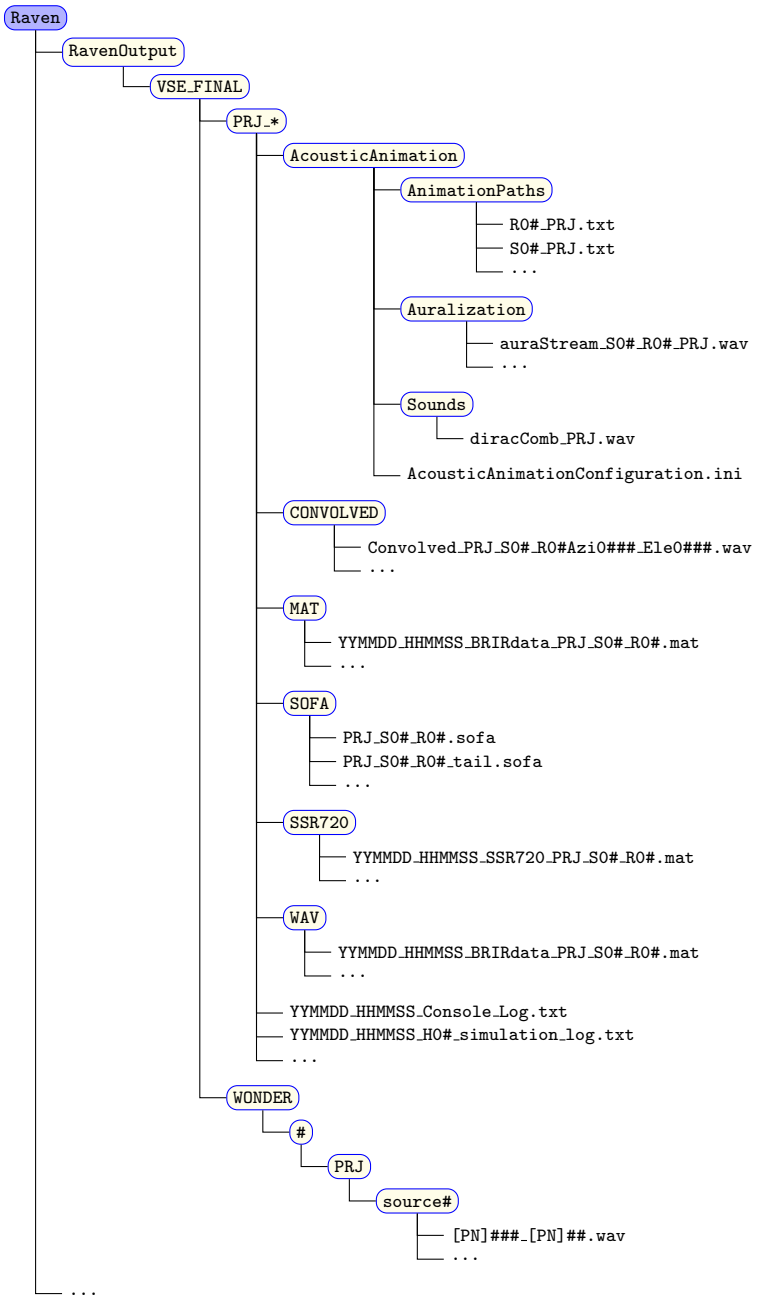


Abbildung C.3.: Ordnerstruktur der Raven Projektdateien: OUTPUT. Der Platzhalter „*“ kann beliebige Namenserweiterungen enthalten. Die Raute (#) dient als Platzhalter für eine Ziffer.

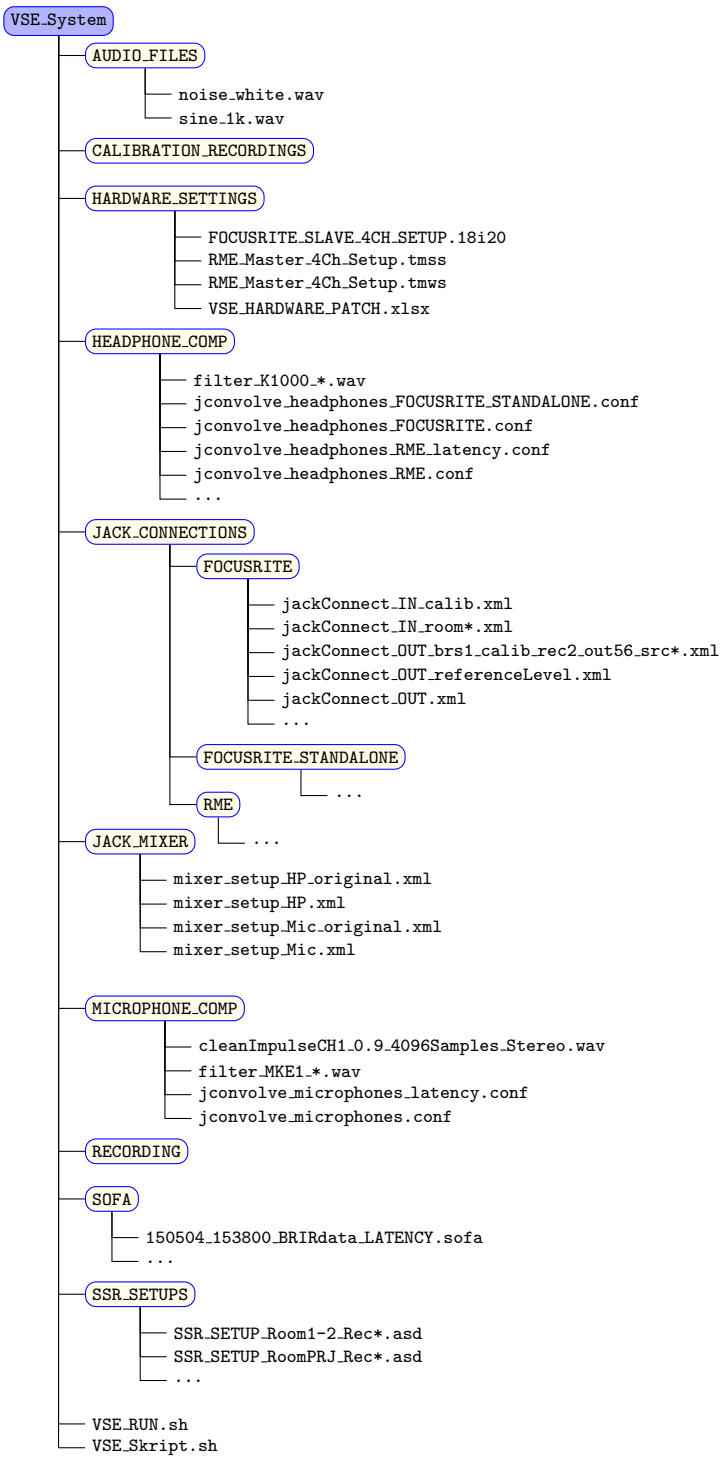


Abbildung C.4.: Ordnerstruktur der VSE Versuchsumgebung. Der Platzhalter „*“ kann beliebige Namenserweiterungen enthalten.

D. Plots und Bildschirmfotos

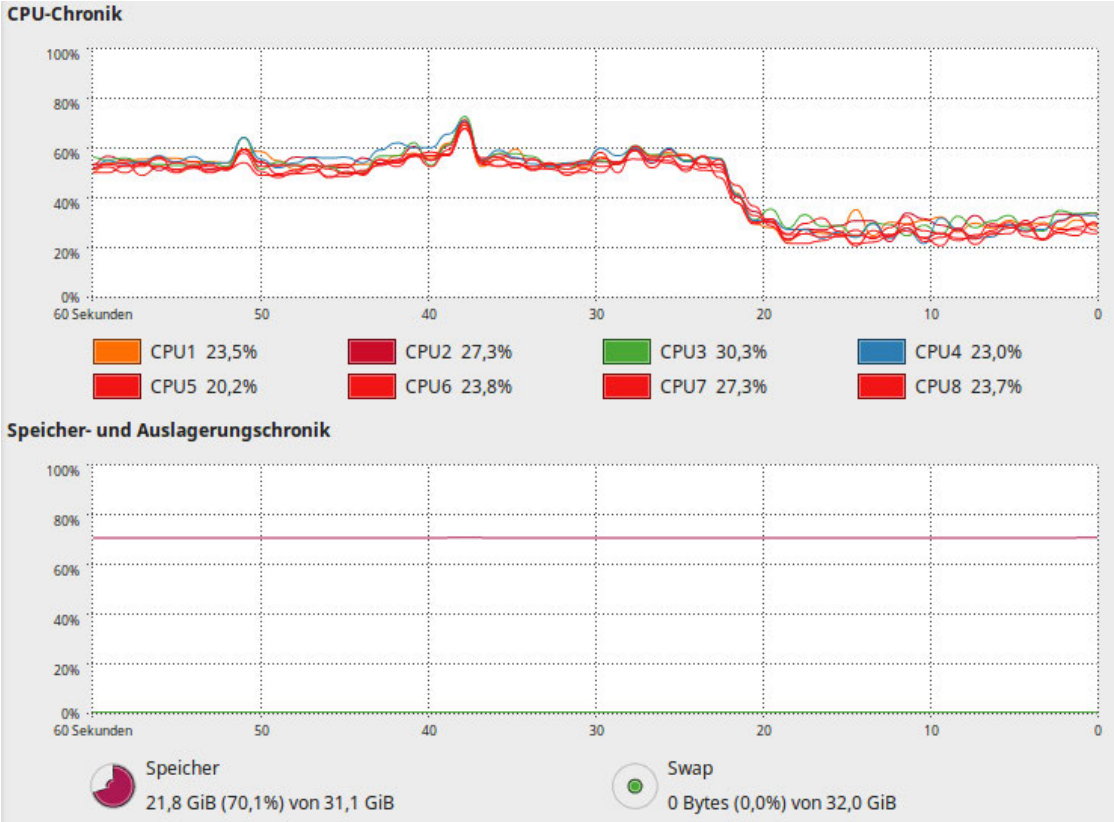


Abbildung D.1.: Auslastung eines Computers während alle benötigten Softwarekomponenten laufen und ein typischer Satz BRIRs für zwei Empfänger und vier Räume in den Speicher geladen wurde. Zum Zeitpunkt t=20 s waren alle Komponenten geladen und die JACK-Audio-Verbindungen hergestellt.

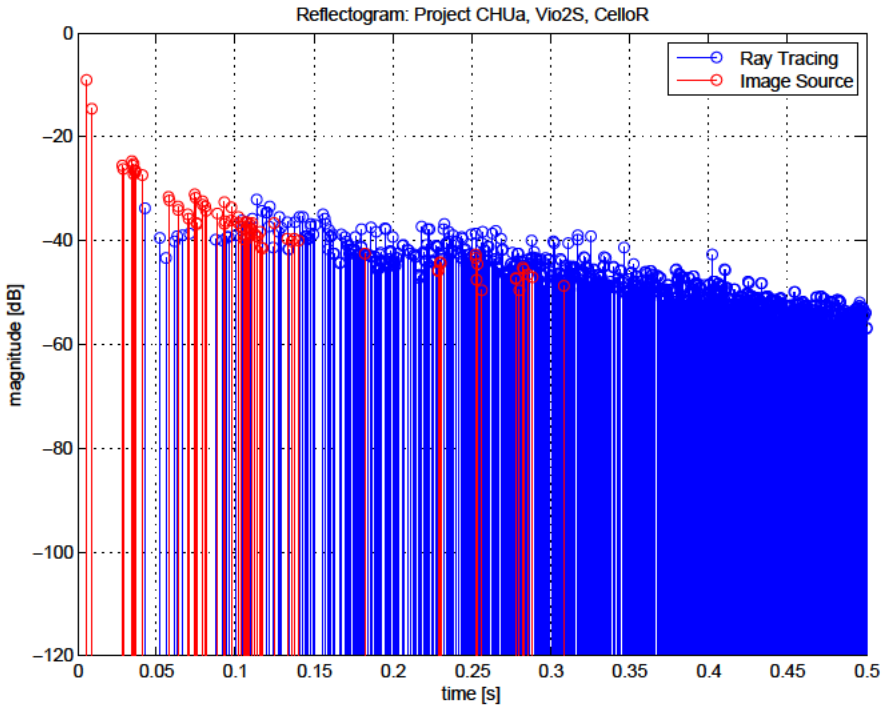


Abbildung D.2.: Reflektogramm für eine Sender/Empfänger-Kombination im Raum CHUa. Rot dargestellt sind die durch die Spiegelquellenmethode berechneten Reflexionen. Blau dargestellt sind die mithilfe des Ray Tracings berechneten diffusen Reflexionen höherer Ordnung.

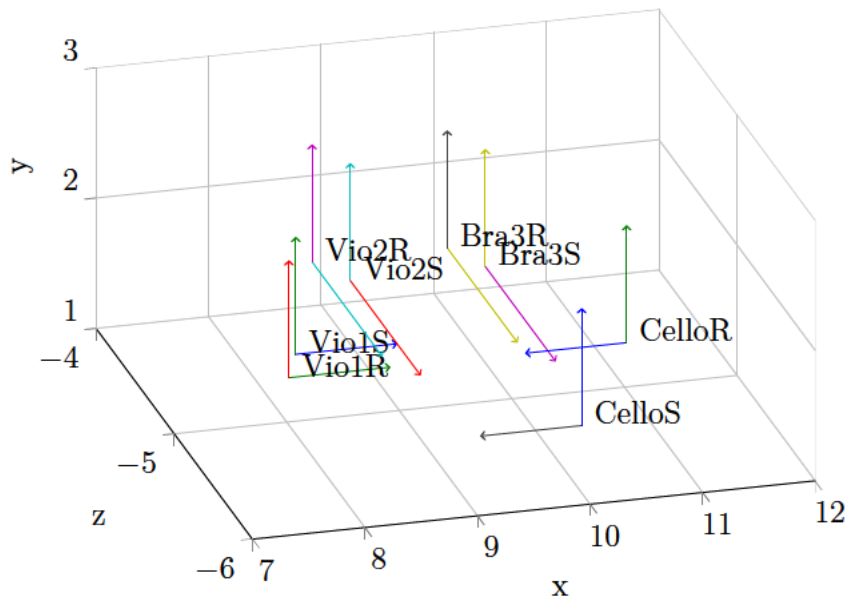


Abbildung D.3.: Visualisierung der zum Modell absoluten Quell- und Empfängerpositionen durch die Funktion *VSE_plotPositions.m*.

E. Hardware-Patchplan

Input					Output				
CH	Typ	Anschluss	Funktion	Inventarnr.	CH	Typ	Anschluss	Funktion	Inventarnr.
1	Line IN	Klinke Sym.	-		1	Line Out	XLR	SSR 1 L	T-Labs
2	Line IN	Klinke Sym.	-		2	Line Out	XLR	SSR 1 R	
3	Line IN	Klinke Sym.	-		3	Line Out	Klinke Sym.	SSR 2 L	2377
4	Line IN	Klinke Sym.	-		4	Line Out	Klinke Sym.	SSR 2 R	
5	Line IN	Klinke Sym.	-		5	Line Out	Klinke Sym.	Kopfhoerer Kalibrierung L	2377
6	Line IN	Klinke Sym.	-		6	Line Out	Klinke Sym.	Kopfhoerer Kalibrierung R	
7	Line IN	Klinke Sym.	-		7	Line Out	Klinke Sym.	-	
8	Line IN	Klinke Sym.	-		8	Line Out	Klinke Sym.	Abhöre Inputs	
9	Mic/Line IN	XLR/Klinke	Mic 1	2381	9	Phones 1 L	St. Klinke	Abhöre SSR 1	
10	Mic/Line IN	XLR/Klinke	Mic 2	2382	10	Phones 1 R			
11	Mic/Line IN	XLR/Klinke	Mic 3	2383	11	Phones 2 L	St. Klinke	Abhöre SSR 2	
12	Mic/Line IN	XLR/Klinke	Mic 4	2384	12	Phones 2 R			
13	AES/EBU 1	XLR	-		13	AES/EBU 1	XLR	-	
14	AES/EBU 2		-		14	AES/EBU 2		-	
15	ADAT 1.1	LWL	-		15	ADAT 1.1	LWL	Direct Out Mic 1	
16	ADAT 1.2		-		16	ADAT 1.2		Direct Out Mic 2	
17	ADAT 1.3		-		17	ADAT 1.3		Direct Out Mic 3	
18	ADAT 1.4		-		18	ADAT 1.4		Direct Out Mic 4	
19	ADAT 1.5		-		19	ADAT 1.5		-	
20	ADAT 1.6		-		20	ADAT 1.6		-	
21	ADAT 1.7		-		21	ADAT 1.7		-	
22	ADAT 1.8		-		22	ADAT 1.8		-	
23	ADAT 2.1	LWL	-		23	ADAT 2.1	LWL	-	
24	ADAT 2.2		-		24	ADAT 2.2		-	
25	ADAT 2.3		-		25	ADAT 2.3		-	
26	ADAT 2.4		-		26	ADAT 2.4		-	
27	ADAT 2.5		-		27	ADAT 2.5		-	
28	ADAT 2.6		-		28	ADAT 2.6		-	
29	ADAT 2.7		-		29	ADAT 2.7		-	
30	ADAT 2.8		-		30	ADAT 2.8		-	

Tabelle E.1.: Patchplan für den Systemaufbau mit der Soundkarte RME Fireface UFX

Input					Output				
CH	Typ	Anschluss	Funktion	Inventarnr.	CH	Typ	Anschluss	Funktion	Inventarnr.
1	Mic IN	XLR	Standalone Mic 1		1	Line Out	Klinke Sym.	SSR 1 L	2389
2	Mic IN	XLR	Standalone Mic 2		2	Line Out	Klinke Sym.	SSR 1 R	
3	Mic IN	XLR	Standalone Mic 3		3	Line Out	Klinke Sym.	SSR 2 L	2376
4	Mic IN	XLR	Standalone Mic 4		4	Line Out	Klinke Sym.	SSR 2 R	
5	Mic IN	XLR	-		5	Line Out	Klinke Sym.	Kopfhörer Kalibrierung L	
6	Mic IN	XLR	-		6	Line Out	Klinke Sym.	Kopfhörer Kalibrierung R	
7	Mic IN	XLR	-		7	Line Out/ Phones 1 L	Klinke	Abhöre SSR 1	
8	Mic IN	XLR	-		8	Line Out/ Phones 1 R	Klinke		
9	SPDIF	Cinch	-		9	Line Out/ Phones 2 L	Klinke	Abhöre SSR 2	
10	SPDIF	Cinch	-		10	Line Out/ Phones 2 R	Klinke		
11	ADAT 1	LWL	IN Mic 1		11	(nicht verbunden via Focusrite Treiber Routing)			
12	ADAT 2		IN Mic 2		12				
13	ADAT 3		IN Mic 3		13				
14	ADAT 4		IN Mic 4		14				
15	ADAT 5		-		15				
16	ADAT 6		-		16				
17	ADAT 7		-		17				
18	ADAT 8		-		18				
					19				
					20				

Tabelle E.2.: Patchplan für den Systemaufbau mit der Soundkarte Focusrite Scarlett 18i20

F. Fotos

(a) Lautsprecher und Normalmikrofon in Messaufstellung.

(b) Normalmikrofon Behringer ECM-8000 mit befestigtem DUT Sennheiser MKE 1.

(c) Normalmikrofon mit befestigtem DUT.

Abbildung F.1.: Kalibrierung der Miniaturkondensatormikrofone.

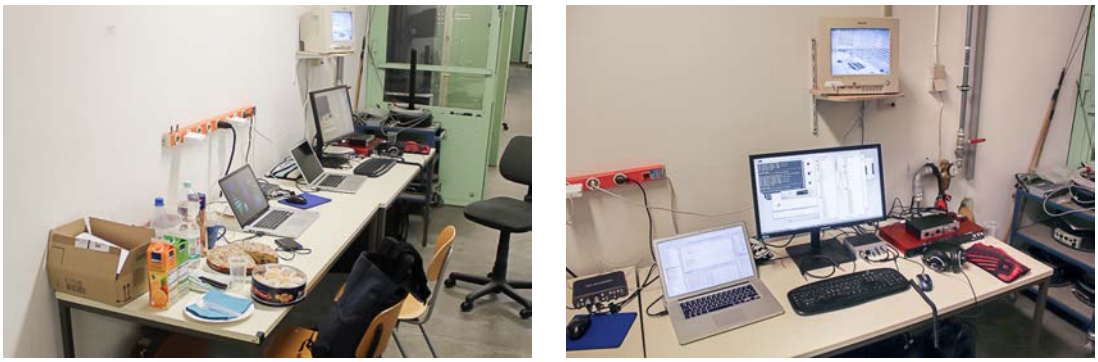


Abbildung F.2.: Kontrollraum mit Rendering-PCs, Messsystem, Monitor mit Live-Bild aus dem RAR und Verpflegung für die Probanden.

Abbildung F.3.: Musiker während des Versuchs im RAR.

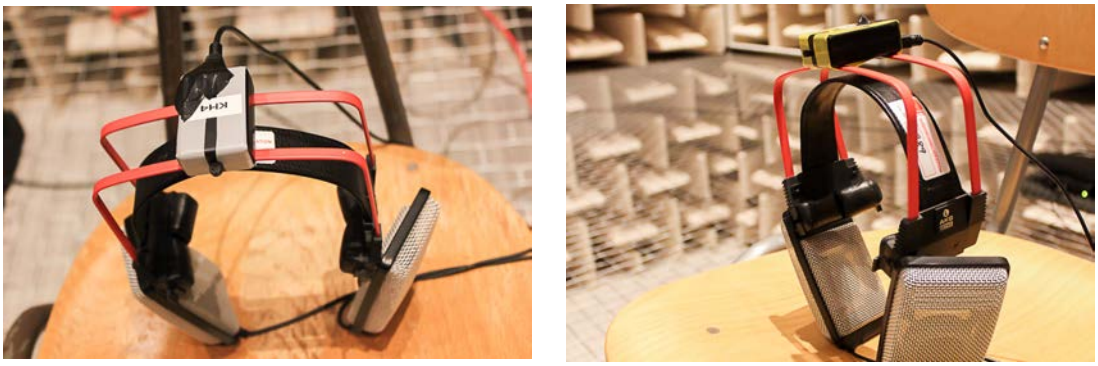


Abbildung F.4.: Extra-aurale Kopfhörer AKG K1000 mit angebautem Headtracker.

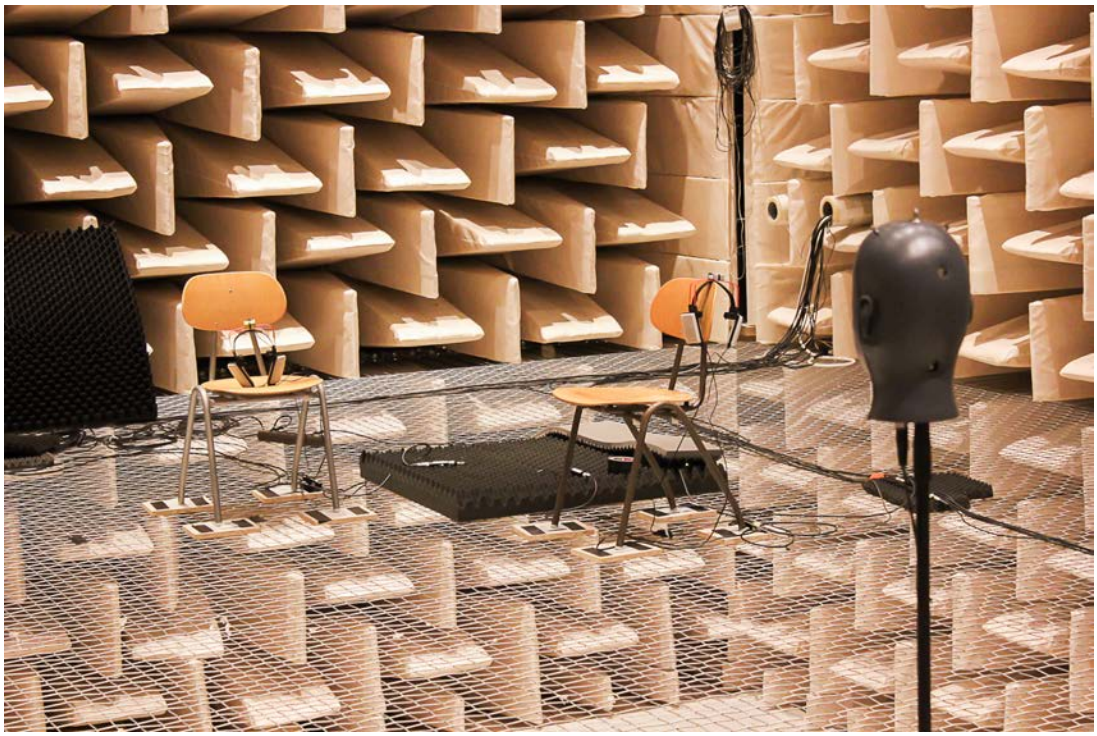


Abbildung F.5.: Kunstkopf für die Pegelkalibrierung mit Blick auf den Aufbau für die Anordnung der Musiker.

G. Datenträger

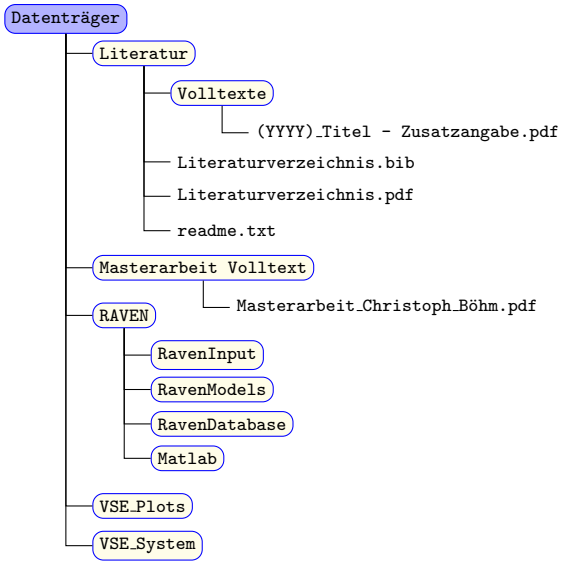


Abbildung G.1.: Inhalt des beiliegenden Datenträgers. Alle im Verlauf dieser Arbeit besprochenen Skripte und Dateien sind enthalten. Die berechneten BRIR-Datensätze und SOFA-Dateien des VSE-Systems können aufgrund ihrer Dateigröße nicht auf dem Datenträger gespeichert werden und liegen dem Fachgebiet auf den für den Versuch verwendeten Computern vor.