Konzeption und Implementierung einer mobilen App für geolokalisierte dynamische Binauralsynthese

Institut für Sprache und Kommunikation
Fachgebiet Audiokommunikation

Abschlussarbeit

zur Erlangung des akademischen Grades
Magister Artium

vorgelegt von

Markus Hädrich

[redacted]

[redacted]

Datum: 9. September 2015

Erstgutachter: Prof. Dr. Stefan Weinzierl

Zweitgutachter: Dr. rer. nat. Alexander Lindau

Fakultät |
Geisteswissenschaften

Prüfungsausschüsse

Studiendekan und Vorsitzender

Prof. Dr. Friedrich Steinle

Eidesstattliche Erklärung

Ist jeder an der TU Berlin verfassten schriftlichen Arbeit eigenhändig unterzeichnet beizufügen!

Hiermit erkläre ich an Eides statt gegenüber der Fakultät I der Technischen Universität Berlin, dass die vorliegende, dieser Erklärung angefügte Arbeit selbstständig und nur unter Zuhilfenahme der im Literaturverzeichnis genannten Quellen und Hilfsmittel angefertigt wurde. Alle Stellen der Arbeit, die anderen Werken dem Wortlaut oder dem Sinn nach entnommen wurden, sind kenntlich gemacht. Ich reiche die Arbeit erstmals als Prüfungsleistung ein. Ich versichere, dass diese Arbeit oder wesentliche Teile dieser Arbeit nicht bereits dem Leistungserwerb in einer anderen Lehrveranstaltung zugrunde lagen.

Titel der schriftlichen Arbeit

KONZEPTION UND IMPLEMENTIERUNG

EINER MOBILEN APP FÜR GEOLOKALISIERTE DYNAMISCHE BINAURALSYNTHESE

VerfasserIn/VerfasserInnen*

Name Hädrich Vorname Markus [redacted]

Betreuende/r DozentIn

Name Weinzierl Vorname Stefan, Prof. Dr.

Lindau Alexander, Dr.

Mit meiner Unterschrift bestätige ich, dass ich über fachübliche Zitierregeln unterrichtet worden bin und verstanden habe. Die im betroffenen Fachgebiet üblichen Zitiervorschriften sind eingehalten worden.

Eine Überprüfung der Arbeit auf Plagiate mithilfe elektronischer Hilfsmittel darf vorgenommen werden.

Berlin, 10.09.2015

Ort, Datum Unterschrift**

*Bei Gruppenarbeiten sind die Unterschriften aller VerfasserInnen erforderlich.
**Durch die Unterschrift bürgen Sie für den vollumfänglichen Inhalt der Endversion dieser schriftlichen Arbeit.

Danksagungen

Mein großer Dank gilt meiner Familie für ihre Unterstützung meines Studiums, Frau Nase für ihren geduldigen Rat bei Tag und Nacht, plaetzchen für seine freundlichen Auskünfte in Programmierfragen und Amaranta Paganelli für das farbenfrohe grafische Design von App-Icons und -Hintergrund. Nicht zuletzt möchte ich Herrn Prof. Dr. Stefan Weinzierl und Herrn Dr. Alexander Lindau für ihre Betreuung danken.

Abstract

Powerful smartphones that continuously gather data via a plethora of sensors have made computing ubiquitous. They have become a possible platform for Augmented Reality Systems as well as Virtual Reality Systems. This thesis designs and implements a mobile app for geo-localized Dynamic Binaural Synthesis on iOS with Pure Data. The scope of the augmentation is limited to audio. The app auralizes a static, virtual sound source — a so-called sound spot — according to the geographical location of the user and his viewing direction. The user can place these sound spots arbitrarily on a map. He can modify their acoustic parameters. The implementation has found a reasonably capable trade-off between hardware and software ressources and a cost-effective but acoustically plausible auralization. Besides accuracy, continuity of the accurate geographical locating has emerged as a challenge.

Kurzfassung

Rechnerallgegenwärtigkeit in Form leistungsstarker Smartphones, die durch eine vielfältige Sensorik fortlaufend Umgebungsdaten sammeln, hat diese zu einer möglichen Plattform für Virtual- und Augmented-Reality-Anwendungen werden lassen. Mit dem Ziel einer rein akustischen Augmentierung wurde innerhalb dieser Arbeit unter iOS eine mobile Augmented-Audio-App für geolokalisierte dynamische Binauralsynthese in Pure Data entwickelt. Diese auralisiert eine sich innerhalb eines sog. Soundspots ausbreitende, ruhende virtuelle Schallquelle in Abhängigkeit von der geografischen Position und der Blickrichtung. Die App ermöglicht es dem Nutzer oder der Nutzerin solche Soundspots geografisch arbiträr zu positionieren und deren akustische Parameter zu modifizieren. Bei der Implementierung wurde ein lauffähiger Kompromiss aus Hard- und Softwareressourcen und einer im Aufwand reduzierten, jedoch akustisch plausiblen Auralisation gefunden. Die dabei zu Tage getretenen Herausforderungen betreffen neben der Genauigkeit vor allem die Stetigkeit der geografischen Ortung.

Inhaltsverzeichnis

Kurzfassung/Abstract **I**

Verzeichnisse **IV**
Abbildungsverzeichnis IV
Abkürzungsverzeichnis V
Formelverzeichnis VII
Symbolverzeichnis VIII
Tabellenverzeichnis X
Quellcodeverzeichnis XI

1 Einleitung **1**

2 Stand der Forschung: Räumliches Hören **3**
2.1 Physikalische Cues 3
2.1.1 Monoaurale Merkmale 4
2.1.2 Binaurale Merkmale 4
2.2 Psychoakustische Merkmale 5
2.2.1 Richtungshören 5
2.2.2 Entfernungsabschätzung 7
2.2.3 Im-Kopf-Lokalisation 8
2.2.4 Physiologische Auswertung binauraler Ohrsignale 9
2.3 Akustische Einflüsse von Umweltfaktoren 11
2.3.1 Meteorologische Cues 12
2.3.2 Einflüsse von Boden und Terrain 15

3 Stand der Technik: Mobile Dynamische Binauralsynthese **21**
3.1 Systemtheoretische Grundlagen 21
3.1.1 Schnelle Fourier-Transformation 21
3.1.2 Diskrete und Schnelle Faltung 25
3.2 Dynamische Binauralsynthese 27
3.2.1 Akustische Richtungssimulation mit HRTFs 28
3.2.2 Akustische Raumsimulation mit (B)RIRs 31
3.2.3 HpTF-Kompensation 31
3.2.4 Headtracking 33
3.3 Augmented Reality 34

4 Methoden: Die App *Binaural-Sound-Map* **36**
4.1 Konzept: Idee, Features und Umsetzung 36
4.2 Werkzeuge: Plattform, Hardware und Programmiersprachen 37
4.3 Sensorbasiertes smartphoneinternes Head- und Positionstracking 38
4.4 *Binaural Soundmap* – Model-View-Controller 48

4.5 Dynamische Datenakquise und -verarbeitung 55
4.5.1 Geodaten 55
4.5.2 Orientierungsdaten 56
4.6 Signaltechnische Simulation psychoakustischer und physikalischer Cues mit Pure Data 59
4.6.1 Simulation der Schallausbreitung im Halbfreifeld durch Spiegelschallquellenmodell 1. Ordnung 60
4.6.2 Einfügen räumlicher Cues durch Faltung mit RIRs 62
4.6.3 Einfügen binauraler Cues durch Faltung mit HRTFs 63
4.6.4 Verbesserung der binauralen Abbildung durch approximierte HpTF-Kompensation 64
4.6.5 Enhancements 65

5 Technische Leistungsbetrachtung 67

6 Fazit 69

7 Ausblick 71

Literatur 73
Normen 73
Publikationen 73
Internetquellen 79

Anhang 85

Abbildungsverzeichnis

1 Kopfbezogenes Koordinatensystem . . . 3
2 Cone of Confusion . . . 7
3 Dissipationsdämpfung der Luft . . . 15
4 Spiegelschallquellenmodell . . . 16
5 Winkelabhängiger Absorptionsgrad . . . 20
6 Radix-2-FFT . . . 24
7 HRIR und HRTF . . . 30
8 Realitäts-Virtualitäts-Kontinuum . . . 35
9 Qualcomm WTR1605L/AKM AK8963 . . . 39
10 Orientierung des Teslameters nach der Nord-Ost-Unten-Konvention . . . 40
11 Magnetfeldmessung . . . 41
12 Orientierung des Accelerometers . . . 42
13 Sensortech BMA220/STMicroelectronics B329 . . . 43
14 Orientierung des Gyroskopes . . . 44
15 Ortungstest am geodätischen Referenzpunkt . . . 46
16 Model-View-Controller (MVC) . . . 48
17 Entitäten (ET) *Soundwalk* und *Soundspot* mit ihren Attributen (AT) . . . 49
18 Binaural Soundmap – GUI . . . 50
19 User-Icon . . . 52
20 Strukturelle Übersicht der *Binaural Soundmap* . . . 53
21 Positions- und Head-Tracking . . . 55
22 Audiosynthese in Pure Data . . . 59
23 Geometrische und dissipationsbedingte Dämpfung . . . 61
24 Elevation der Bodenreflexion . . . 62
25 Verhallung durch Faltung mit OIR . . . 63
26 Approximierte HpTF-Kompensation . . . 65
27 Winkelabhängiger EQ . . . 66
28 CPU-Nutzung im Vollbetrieb . . . 67
29 App-Icon . . . 69

Abkürzungsverzeichnis

A-GPS Unterstütztes Globales Ortsbestimmungssystem (Assisted Global Positioning System) . . . 38
AR Augmentierte Realität (Augmented Reality) . . . 21
ARA Akustisch Augmentierte Realität (Augmented Reality Audio) . . . 35
ARS System für Augmentierte Realität (Augmented Reality System) . . . 34
API Programmierschnittstelle (Application Programming Interface) . . . 37
AV Augmentierte Virtualität (Augmented Virtuality) . . . 34
BS Binauralsynthese (Binaural Synthesis) . . . 1
BRIR Binaurale Raumimpulsantwort (Binaural Room Impulse Response) . . . 28
BF Binauraler Filter (Binaural Filter) . . . 1
BDS Großer Bär (BeiDou) . . . 38
COC Kegel unsicherer Lokalisation (Cone of Confusion) . . . 6
CBF Rückruf-Funktion (Call Back Function) . . . 48
DL-Lemma Danielson-Lanczos Lemma . . . 23
DBS Dynamische Binauralsynthese (Dynamic Binaural Synthesis) . . . 1
DFT Diskrete Fouriertransformation (Diskrete Fourier Transform) . . . 21
DSP Digitaler Signalprozessor (Digital Signal Processor) . . . 25
DH Kunstkopf (Dummy-Head) . . . 28
GDGPS Globales Ortsbestimmungssystem mit Differentialsignal (Differential Global Positioning System)
JND Unterschiedsschwelle (Just Noticable Difference) . . . 4
FBA Vorne-Hinten Ambiguität (Front-Back Ambiguity) . . . 7
FG Frequenzgruppe . . . 6
FT Fouriertransformation (Fourier Transform) . . . 21
FFT Schnelle Fourier Transformation (Fast Fourier Transform) . . . 21
FIR System mit endlicher Impulsantwort (Finite Impulse Response) . . . 27
GUI Grafische Benutzerschnittstelle (Graphical User Interface) . . . 36
GPS Globales Ortsbestimmungssystem (Global Positioning System) . . . 38
GSM Globales System für Mobile Kommunikation (Global System for Mobile Communications) . . . 39
GNSS Globales Navigationssatellitensystem (Global Navigation Satellite System) . . . 38
GLONASS Globales satellitengestütztes Navigationssystem (GLObal NAvigation Satellite System) . . . 38
HCI Mensch-Computer-Schnittstelle (Human Computer Interface) . . . 35
HRIR Kopfbezogene Impulsantwort (Head Related Impulse Response) . . . 29
HRTF Außenohrübertragungsfunktion (Head Related Transfer Function) . . . 1
HMD Videobrille (Head Mounted Display) . . . 35
HpTF Kopfhörerübertragungsfunktion (Headphone Transfer Function) . . . 32
IDFT Umkehrfunktion der Diskreten Fouriertransformation (Inverse Diskrete Fourier Transform) . . . 22

IFFT Umkehrfunktion der Schnellen Fouriertransformation (Inverse Fast Fourier Transform) . . . 25
IHL Im-Kopf-Lokalisation (Inside-the-Head Locatedness) . . . 8
IK Interaurale Kohärenz . . . 11
ILD Interaurale Pegeldifferenz (Interaural Level Difference) . . . 4
IPD Interaurale Phasendifferenz (Interaural Phase Difference) . . . 5
ITD Interaurale Laufzeitdifferenz (Interaural Time Difference) . . . 4
ITDG Anfangszeitlücke (Initial Time Delay Gap) . . . 4
IMU Inertialsensor (Inertial Measurement Unit) . . . 1
IR Impulsantwort (Impulse Response) . . . 25
HP Kopfhörer (Headphone) . . . 32
LsTF Lautsprecherübertragungsfunktion (Loudspeaker Transfer Function) . . . 32
LCS Ortsgebundende Datenverarbeitung und Dienste (Location-based Computing and Services) . . . 36
LTI-System Lineares Zeitinvariantes System (Linear Time Invariant System) . . . 27
LAA Ortsgebundenes Augmented Audio (Location-based Augmented Audio) . . . 36
LSB Niedrigwertigstes Bit (Least Significant Bit) . . . 39
MR Gemischte Realität (Mixed Reality) . . . 34
MARA Mobile Akustisch Augmentierte Realität (Mobile Augmented Reality Audio) . . . 1
MARS System für Mobile Augmentierte Realität (Mobile Augmented Reality System) . . . 1
MEMS Mikroelektromechanisches System (Microelectromechanical System) . . . 42
MLS Maximallängenfolge (Maximal Length Sequence) . . . 63
MVC Modell-Präsentation-Steuerung (Model View Controller) . . . 48
NED Nord-Ost-Unten-Konvention (North-East-Down-Convention) . . . 40
OLA Overlap-Add-Verfahren . . . 26
OLS Overlap-Save-Verfahren . . . 26
OHL Außer-Kopf-Lokalisation (Outside-the-Head Localization) . . . 9
OIR Outdoor-Impulsantwort (Outdoor Impulse Response) . . . 1
SPS Standard Ortsbestimmungsservice (Standard Positioning Service) . . . 39
UX Anwendererlebnis (User Experience) . . . 50
UC Rechnerallgegenwärtigkeit (Ubiquitous Computing) . . . 1
VR Virtuelle Realität (Virtual Reality) . . . 34
VRC (Virtual Reality continuum) . . . 34
Virtualitäts-Realitäts-Kontinuum

Formelverzeichnis

2.1 Dopplereffekt bei ruhender Schallquelle . . . 4
2.2 Interaurale Laufzeitdifferenz . . . 5
2.3 Zusammenhang von realer und erwarteter Schallereignisentfernung . . . 8
2.4 Normierte Kreuzkorrelation . . . 10
2.5 Interauraler Kohärenzgrad . . . 10
2.6 Interaurale Signalunterschiede . . . 10
2.7 Interaurale Pegeldifferenz . . . 10
2.8 Interaurale Phasendifferenz . . . 10
2.9 Interaurale Gruppenlaufzeit . . . 11
2.10 Schalldämpfung im Freien . . . 11
2.11 Schallgeschwindigkeit in Abhängigkeit von der Temperatur . . . 12
2.12 Schallgeschwindigkeit in Abhängigkeit von der Temperatur (vereinfacht) . . . 12
2.13 Schallausbreitung im Freifeld . . . 13
2.14 Absorptionsgrad der Luft . . . 14
2.15 Diffusfeld-Reflexionsfaktor . . . 18
2.16 Winkelabhängiger Reflexionsfaktor . . . 18
2.17 Bodenimpedanz nach Delany, Bazley und Miki . . . 19
2.18 Winkelabhängiger Absorptionsgrad . . . 19
3.1 Kontinuierliches Fourier-Transformationspaar . . . 21
3.2 Diskretes Fourier-Transformationspaar . . . 22
3.3 Diskretes Betrags- und Phasenspektrum . . . 22
3.4 Diskrete Fourieranalyse . . . 23
3.5 Diskrete Faltung im Zeitbereich . . . 25
3.6 Diskrete Faltung im Frequenzbereich . . . 25
3.7 Overlap-Add Segmentierung . . . 26
3.8 Overlap-Add Superposition . . . 26
3.9 Bestimmung der Außenohrübertragungsfunktion . . . 29
3.10 Reduzierung des HRTF-Datensatzes . . . 31
3.11 Bedingungen an HpTF-Entzerrfilter . . . 32
3.12 HpTF-Kompensationsfilter . . . 32
4.1 Niederwertigstes Bit . . . 40
4.2 Beschleunigungsvektor . . . 42
4.3 Beschleunigungsmessung . . . 42
4.4 Distanz zwischen zwei geografischen Koordinaten . . . 46
4.5 1/r-Gesetz . . . 60
4.6 Biquad Filter . . . 61
4.7 HRTF-Interpolation, Schritt 1 . . . 64
4.8 HRTF-Interpolation, Schritt 2 . . . 64

Symbolverzeichnis

1. Kennzeichnung durch Art der Schreibweise

$y(t),\ h(t)$, etc.	zeitkontinuierliche Werte
$y(n),\ h(n)$, etc.	zeitdiskrete Werte

2. Symbole

Abkürzung	*Variable*	*Einheit*
A	Dämpfung	dB
α	Absorptionsgrad	
c	Schallgeschwindigkeit	m/s
δ	Schichtdicke	m
DI	Richtwirkungsindex	dB
E	Sättigungsdruck	hPa
f	Frequenz, Funktion	Hz = 1/s
g	Fallbeschleunigung	m/s^2
h	Höhe	m
H	Übertragungsfunktion	
k	Konstante, Laufvariable	
L	Schallpegel	dB
N	Abtaststelle	
p	Druck	Pa
ψ	Schalleinfallswinkel	°
ρ	Dichte	g/cm^3 kg/m^3
r	Radius, Abstand Sender/Empfänger	m
R	Reflexionsgrad	
σ	Strömungswiderstand	kN·s/m^4
T	Zeitkonstante/Periode	s
ϑ	- Temperatur	°C
	- absolute Temperatur	K
θ	Elevationswinkel, Schalleinfallswinkel	°
φ	Azimuthwinkel	°
ω	Winkelgeschwindigkeit	rad/s
Q	Richtwirkungsfaktor	
v	Geschwindigkeit	m/s
Z	Schallkennimpedanz	N·s/m^3

3. Indizes

Index	*Bezeichnung, Äquivalent*	*Einheit*
a_{diff}	Diffusfeldabsorptionsgrad	
c_0	Referenzschallgeschwindigkeit	m/s
δ_K	zeitdiskreter Einheitsimpuls, Kronecker-Delta	
F_r	Relaxationsfrequenz	Hz
h_s	Höhe über dem Meeresspiegel, Senderhöhe	m
h_r	Empfängerhöhe	m
H_r	relative Luftfeuchtigkeit	%
ITD_{max}	maximale interaurale Laufzeitdifferenz	ms
I_P	Impulsantwort des aktuellen Blocks	
I_{P_0}	Impulsantwort des vorangegangenen Blocks	
r'	wahrgenommener Abstand Sender/Empfänger	m
r_{diff}	Diffusfeldreflexionsfaktor	
T_A	Abtastperiode	ms
T_0	Referenztemperatur	°
τ_{ph}	Phasenlaufzeit	ms
τ_{gr}	Gruppenlaufzeit	ms
ω_0	Abtastkreisfrequenz, $2\pi f_0 = \frac{2\pi}{T_0}$	s^{-1}
R_{norm}	normierte Kreuzkorrelation	
T_Q	Quantum Tesla Level	μT
W_N	Drehfaktor, $e^{-j\frac{2\pi}{N}}$	
Z_0	Schallkennimpedanz der Luft, $\rho_0 \cdot c_0$	Pa·s/m^3

Tabellenverzeichnis

1 Strömungswiderstand einiger Bodenoberflächen 19
2 Akquise der Positionsdaten 38
3 Accelerometer BMA220 43
4 Ortungstest am geodätischen Referenzpunkt 47

Quellcodeverzeichnis

1 Core Location - Teslameter Kalibrierung 45
2 Core Location - Genauigkeitskonstanten der Ortungsdienste 45
3 Core Data - NSFetchedResultsController 54
4 Core Location - Distanzbestimmung 56
5 Core Motion - CMPedometer 56
6 Richtung des Soundspots bezüglich der User-Position 57
7 Elevationsbestimmung 57
8 NSOperationQueuePriority 58
9 Ortungsgenauigkeit des iPhone 5s am geodätischen Referenzpunkt Berlin Mitte . . 85
10 Latenz Azimuthberechnung 86
11 Latenz der Elevationsberechnung 86
12 Systemlatenz der App 87
13 Latenz des PD-Objektes `earplug~` 87
14 Latenz des PD-Objektes `convolve~` 87

1 Einleitung

Bei Mobile-Augmented-Reality treffen sich Mark Weiser mit seiner Vision einer *Rechnerallgegenwärtigkeit* (Ubiquitous Computing, UC) [5] und Ivan E. Sutherland mit seiner Idee einer computergenerierten Erweiterung der menschlichen Wahrnehmung und Realität [6]. Längst über die geerbte Funktion der Telefonie herausgewachsen, avancieren Smartphones zum wichtigsten Gerät bei der privaten Internetnutzung [106] und der mobilen Rezeption von Audioinhalten [107]. Durch die enorme Leistungssteigerung [108], geringe Größe, die Integration von Kamera(s), Mikrofon(en), *Inertialsensoren (Inertial Measurement Units)* (IMUs) und satellitengestützter Navigationssysteme, können moderne Smartphones fortlaufend Umgebungsvariablen, Bewegungen und damit auch Nutzerverhalten erfassen. Diese Informationen, ausgewertet durch eine intelligente Software, können für eine Registrierung augmentierter, computergenerierter Inhalte in die reale Umgebung [7] genutzt werden. Auf diese Weise kann mobiles UC einem *System für Mobile Augmentierte Realität* (Mobile Augmented Reality System, MARS) eine Plattform bieten.
In dieser Arbeit soll ein Teilaspekt von MARS in Form einer Anwendung für *Mobile Akustisch Augmentierte Realität* (Mobile Augmented Reality Audio, MARA) herausgegriffen werden. Die Forderung Azuma's nach einer in diesem Fall rein akustischen Registrierung in 3D [8] gibt die Verwendung eines 3D-Audiosystems vor, das die Kopfbewegungen und die räumliche Position des Nutzers oder der Nutzerin im virtuellen Schallfeld bei der Audiosynthese berücksichtigt. Da es sich bei dem iPhone 5s als gewählte Hardware trotz allem um ein System mit geteilten und begrenzten Ressourcen handelt, soll als Mittel der Auralisation die kopfhörergebundene *Binauralsynthese* (Binaural Synthesis, BS) als eine der „günstigsten" Methoden der virtuellen Akustik zum Einsatz kommen. Hier müssen nur die akustischen Parameter zweier Audiokanäle an jeweils einer Position berechnet werden. Durch deren dynamische Ausrichtung, bei der *Binaurale Filter (Binaural Filters)* (BFs) mit den Bewegungen des Nutzers oder der Nutzerin synchronisiert in Echtzeit ausgetauscht werden, soll eine immersive Audio-Performance erreicht werden.
Auf diese Weise soll eine MARA-Anwendung für geolokalisierte *Dynamische Binauralsynthese* (Dynamic Binaural Synthesis, DBS) in urbanen Außenumgebungen geschaffen werden, die eine plausible akustische Simulation ruhender Punktschallquellen im teilbegrenzten Halbfreifeld ermöglicht. Mit dem Fokus auf der softwareseitigen Umsetzung von Methoden der virtuellen Akustik soll die Akquise der für die DBS notwendigen Bewegungsdaten des Nutzers oder der Nutzerin mit Hilfe der smartphoneinhärenten IMUs und mittels A-GPS erfolgen. Plausible Richtungs- und Rauminformationen sollen durch geeignete signaltechnische Simulationen von Abstands- und Luftdämpfung, Bodenreflexionen, dem dynamischen Austausch von nicht-individuellen *Außenohrübertragungsfunktionen (Head Related Impulse Responses)* (HRTFs) sowie einer Verhallung mit monauralen *Outdoor-Impulsantworten (Outdoor Impulse Responses)* (OIRs) prototypischer Außenumgebungen erreicht werden. Auf diese Art soll die App eine sich innerhalb sog. Soundspots ausbreitende, ruhende virtuelle Schallquelle in Abhängigkeit von der geografischen Position und der Blickrichtung auralisieren.
Zu Beginn sollen in Kapitel 2 die Parameter des räumlichen Hörens betrachtet und aus ihnen Anforderungen an die akustische Simulation abgeleitet werden. Anschließend werden in Kapitel 3 mögliche Methoden zu deren signaltechnischer Umsetzung vorgestellt. Der Hauptteil ab Kapitel 4 beschreibt die Implementierung der App. Hier wird zuerst das Konzept zur Softwareerstellung und

der dazu gewählten Programmiersprachen in Unterkapitel 4.1 vorgestellt. Anschließend wird in Unterkapitel 4.2 die vom Verbund aus Hard- und Software zu erwartenden Leistungen hinsichtlich Bewegungs- und Positionserfassung auf Grundlage einer Recherche relevanter technischer Datenblätter und Software-Referenzen prognostiziert. In Unterkapitel 4.4 folgt ein Überblick über die Architektur der mobilen App, ein detaillierter Einblick in die Akquise der Bewegungsdaten und in die Virtualisierung der akustischen Szene. Abschließend in Kapitel 5 wird die App-Performance einer technischen Leistungsbetrachtung unterworfen. Nach der Zusammenfassung der Arbeitsergebnisse in Kapitel 6 werden in Kapitel 7 mögliche Wege einer zukünftigen Weiterentwicklung vorgeschlagen.

2 Stand der Forschung: Räumliches Hören

Im ersten Abschnitt werden die am räumlichen Hörprozess beteiligten physikalischen Parameter und ihre psychoakustischen Wirkungen betrachtet. Im zweiten Abschnitt wird auf deren wichtigste psychoakustisch wirksamen Phänomene eingegangen.

2.1 Physikalische Cues

Zur räumlichen Lagedarstellung von Schallereignissen nutzt man innerhalb kopfbezogener Ansätze zur Schallfeldbeschreibung meist auch das kopfbezogene Koordinatensystem. Der Betrag des Ortsvektors vom Mittelpunkt der interauralen Achse zum Schallereignisort[1] wird in Abbildung 1 durch den Radius r dargestellt. Die Richtungskoordinaten werden durch die Winkel Azimut φ und Elevation θ gebildet.

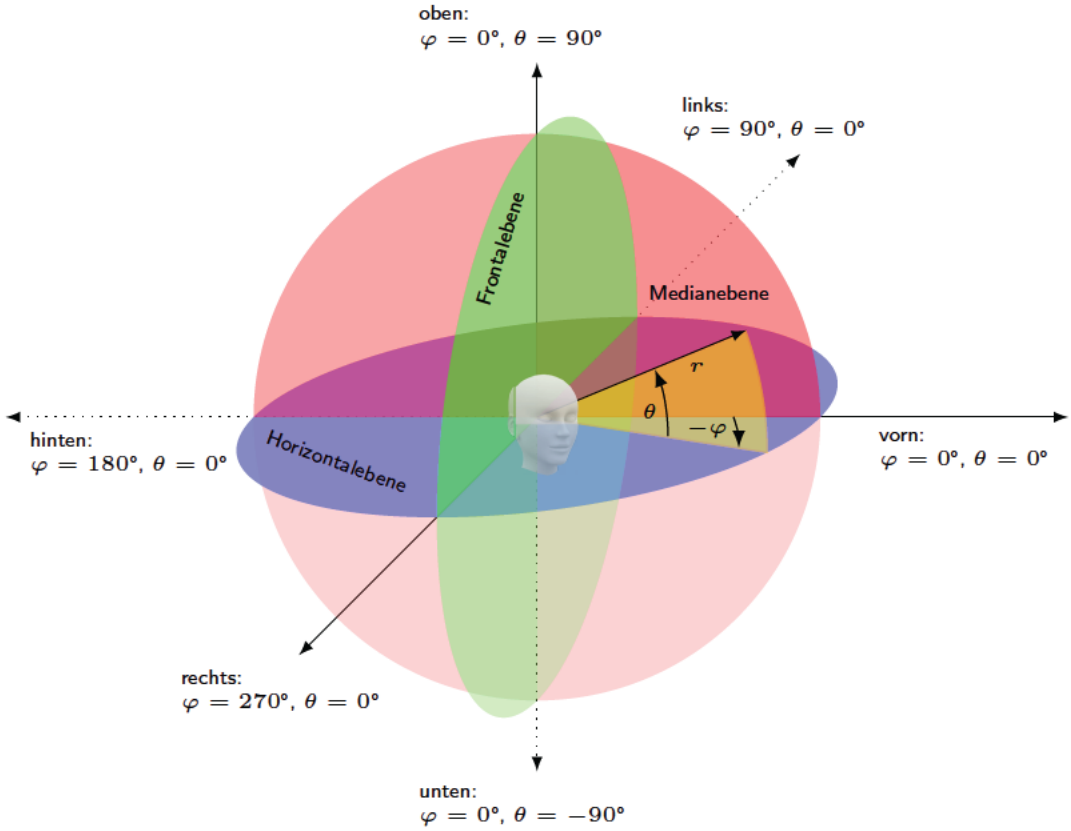


Abb. 1: Kopfbezogenes Koordinatensystem

Grundsätzlich wird zwischen monoauralen und binauralen Merkmalen des räumlichen Hörens unterschieden. Während die Wahrnehmung monoauraler Cues auch mit einohrigem Hören möglich

[1] Schallereignisort $\neq$ Hörereignisort

ist, beruhen die Mechanismen des binauralen Hörens auf der Auswertung nichtkohärenter Anteile der interauralen Ohrsignale.

2.1.1 Monoaurale Merkmale

Die richtungsselektive Filterwirkung von Ohrmuschel, Kopf, Schulter, Torso verursacht auch einohrig wahrnehmbare spektrale Verfärbungen des Ohrsignals. Zu den monoauralen Cues zählt ebenfalls die frequenzunabhängige Pegelabnahme im Freifeld mit 6 dB pro Entfernungsverdopplung nach dem $1/r$-Gesetz und die durch die Dissipation der Luft hervorgerufene Dämpfung hoher Frequenzen. Ein weiteres vom Ohr-Gehirn-System verarbeitetes physikalisches Merkmal ist die sog. *Anfangszeitlücke* (Initial Time Delay Gap, ITDG) zwischen dem ersten Eintreffen des Direktschalls bis zum Eintreffen der ersten Reflexionen.
Bei bewegten Schallsendern und/oder -empfängern kommt es je nach Bewegungsrichtung zu einer Veränderung der wahrgenommenen Tonhöhe aufgrund des *Dopplereffekts*.

Gleichung 2.1: Dopplereffekt, Schallquelle ruhend – Beobachter bewegt [9, S. 52]

Annäherung/Entfernung:

$$f_B = f_S(1 \pm \frac{v}{c})$$

$f_B =$ Frequenz am Beobachtungsort
$f_S =$ Frequenz der Schallquelle
$c =$ Schallgeschwindigkeit: 343 m/s bei $\vartheta = 20$°C
$v =$ Geschwindigkeit des Beobachters

Für den zu implementierenden Fall einer ruhenden Schallquelle und einer angenommenen Fortbewegungsgeschwindigkeit von ≈ 4 km/h für Fußgänger bleibt dieser Effekt unterhalb der *Unterschiedsschwelle* (Just Noticable Difference, JND) für die Tonhöhenwahrnehmung von Sinustönen von $\approx 3,6$ Hz für $f_S < 500$ Hz und relativ konstanten $0,007 f_S$ für höhere Frequenzen. [10]

2.1.2 Binaurale Merkmale

Die Erkenntnis, dass der beim Hörer eintreffende Schall *Interaurale Laufzeitdifferenzen (Interaural Time Differences)* (ITDs) und *Interaurale Pegeldifferenzen (Interaural Level Differences)* (ILDs) beinhaltet und diese vom auditiven System zur Lokalisation herangezogen werden, wird Lord Rayleigh zugeschrieben, der seine Duplex-Theorie um 1907 veröffentlichte. [11] Diese Theorie lieferte die Erklärung für Phänomene der Lokalisation bei seitlichen Auslenkungen, nicht jedoch für die Vorn-Hinten-Richtung. Als ILDs bezeichnet man die durch anatomisch bedingte Abschattungen und Resonanzen an Kopf, Schultern und Torso hervorgerufenen frequenzabhängigen Pegelunterschiede ΔL zwischen beiden Ohrsignalen. Nach dem Huygens-Fesnelschen Prinzip kann eine ebene Wellenfront als Superposition kugelförmiger Elementarwellen [12, S. 666] beschrieben werden.

Fehlt aufgrund eines Hindernisses ein Abschnitt in dieser Wellenfront, kommt es durch den kugelförmigen Ausbreitungscharakter der Elementarwelle zu einer Beugung um das Hindernis herum. Für Wellenlängen λ in der Größenordnung des Kopfdurchmessers stellt dieser ein Hindernis dar, das anfangs Beugungen und mit zunehmender Frequenz Abschattungen durch Reflexionen hervorruft. Als Folge treten mit abnehmender Wellenlänge stärker werdende richtungsabhängige lineare Verzerrungen im Frequenzgang auf.
Die Resonanzen, die in den Gehörmuscheln gebildet werden, erzeugen individualspezifische, richtungsbestimmende Bänder, deren stärkste Anteile bei Breitbandsignalen richtungsbestimmender sind als die tatsächliche Schalleinfallsrichtung[1]. Der Abstand der Gehöreingänge von durchschnittlich $a = 21,5$ cm verursacht durch unterschiedlich lange Signalwege bei seitlichem Schalleinfall zusätzlich ITDs $\Delta\tau$.

Gleichung 2.2: Interaurale Laufzeitdifferenz

$$\Delta t = \frac{a \cdot sin\varphi}{c}, \qquad \text{mit Ohrenabstand } a \approx 21,5 \text{ cm}$$

Die ITDs werden vom Gehör-Gehirn-System auf unterschiedliche Art zur Schallereignisbestimmung herangezogen. Nach Blauert und Braasch [14] nutzt das Gehör-Gehirn-System *Interaurale Phasendifferenzen (Interaural Phase Differences)* (IPDs) $\Delta\tau_{ph}$ für Frequenzen $f < 1,6$ kHz bei denen der Kopfdurchmesser a klein im Verhältnis zur Wellenlänge λ ist. Für höhere Frequenzen sind diese $\Delta\tau_{ph}$ aufgrund der Kopfabmessungen nicht mehr eindeutig. Dementsprechend ergaben Hörversuch nach Seeber [15, S. 13] eine mit der Frequenz wachsende Detektionsschwelle für IPDs mit einem Grenzwert bei $f \approx 1,5$ kHz. Er vermutet einen kognitiven Ausblendmechanismus für vieldeutige Winkelinformationen oberhalb dieses Grenzwertes. Stattdessen verarbeitet das Gehör-Gehirn-System für höhere Frequenzen die auf die Hüllkurve bezogenen Gruppenlaufzeiten $\Delta\tau_{gr}$. Nicht zuletzt gehören binaurale Unterschiede des zeitlichen Verlaufs der in Abschnitt 2.1.1 angeführten ITDG in diese Rubrik.

2.2 Psychoakustische Merkmale

Die psychoakustischen Merkmale des räumlichen Hörens können in Wahrnehmungskategorien wie Geometrie, Timbre, Tonalität, Räumlichkeit, zeitliche Struktur oder Immersion zugeordnet werden, um nur einige zu nennen. Lindau et al. [16] haben diese und weitere Begriffe zu einem Katalog für die Beurteilung der Audio-Qualität von 3D-Audio-Wiedergabesystemen integriert [109]. In den folgenden Abschnitten sollen nur einige der wichtigsten Merkmale herausgegriffen und in ihrer psychoakustischen Wirkung auf räumliche Aspekte wie Richtung und Entfernung hin erläutert werden.

2.2.1 Richtungshören

Die für die Identifikation der Schalleinfallsrichtung nutzbaren monoauralen und binauralen Cues hängen in ihrer Verfügbarkeit und Verwertbarkeit stark von der Schalleinfallsrichtung, der Fre-

[1] Blauertsche Bänder: z. B. Elevationseffekt bei 8 kHz, vgl. Madole und Begault [13, S. 52]

quenz, Signaldauer und -beschaffenheit ab, was sich damit auch in der Lokalisationsunschärfe widerspiegelt. Nach Blauert und Braasch [14] beträgt diese für Auslenkungen auf der Medianebene, bei der weder ITDs noch ILDs auftreten, $\pm 10°$. Im Gegensatz dazu beläuft sich diese für frontale seitliche Auslenkungen aus der Medianebene unter Einbeziehung binauraler Cues auf $\pm 1°$ bis $\pm 4°$ [14, 15]. Das Auftreten von ILDs ist im gesamten hörbaren Frequenzbereich ab 800 Hz mit einer seitlichen Auslenkung des Hörereignisortes verbunden.[1] Mit einem Wertebereich von $\approx 2 \cdot 10^3$ registriert das Gehör bereits ITDs von $10 - 30$ μs bis hin zu $ITD_{max} = 63$ ms bei einem seitlichen Schalleinfall von $\pm 90°$. [14]
Die ITDG ist besonders wichtig für die Richtungslokalisation in halliger Umgebung, da aufgrund des Präzedenz-Effekts allein in dieser Zeit die Lokalisation stattfindet. Für die Richtungsbestimmung (und Lautheitsabschätzung) ist allein die zuerst eintreffende Wellenfront ausschlaggebend. Selbst wenn eine um 20 ± 10 ms verzögerte Reflexion aus einer anderen Richtung eintrifft und einen um bis zu 10 dB höheren Schalldruckpegel aufweist, kommt es durch den *Haas-Effekt* [17] zu keiner Echowahrnehmung. Der Präzedenz-Effekt wird auch oft „Gesetz der ersten Wellenfront“ genannt und bezieht sich auf Reflexionen mit einer Verzögerungszeit von $2 - 50$ ms. Spätere Reflexionen haben keinen Einfluss mehr auf die Lokalisation, da der Hörer ab einem Delay von ca. 40 ms zunehmend zeitlich, u. U. auch räumlich getrennte Hörereignisse wahrnimmt. Diese Echoschwelle ist signalabhängig und liegt bei Musik bei ca. 80 ms [14], was die obere zeitliche Grenze des Geltungsbereiches des Präzedenzeffektes ebenfalls signalabhängig verschiebt. Die einmal lokalisierte Schalleinfallsrichtung bleibt bis zum erneuten Eintreffen von Direktschall erhalten. Ausgenommen ist das reine Diffusfeld: hier verliert sich die Richtungsempfindung nach einiger Zeit. Die Auswertung der Ohrsignale erfolgt in Frequenzgruppen durch Bandpasszerlegung im Innenohr. Die Zerlegung erfolgt bis 500 Hz in 100 Hz - Schritten, für $f > 500$ Hz in Abständen einer kleinen Terz in insgesamt ca. 24 Teilbereiche zu je 1 bark oder 100 mel. Für die Unterscheidung von Schallereignissen werden starke Lautstärkeänderungen in den einzelnen *Frequenzgruppen* (FGs) ausgewertet. Signifikante Änderungen der Schallsignaleigenschaften einer bestehenden Quelle oder das Einsetzen eines neuen Schallsignals werden vom Gehör-Gehirn-System als starke Änderung in einer oder mehreren FGs registriert.
Eine Schallquelle mit kleinen Abmessungen im Verhältnis zur Wellenlänge kann als Punktschallquelle mit $L = 10 \cdot log_{10}(4\pi \cdot r^2/1m^2)$ mit kugelförmiger Ausbreitung betrachtet werden, solange für deren Ausdehnung l und dem Abstand s zum Empfänger $l \leq 0.7 \cdot s$ gilt. [1] Für eine hinreichend praxistaugliche Genauigkeit muss $r > 1,5 \cdot l$ und $r > \lambda_{max}$ erfüllt sein. [18, S. 57]
Die Absorptionseigenschaften des Untergrundes bewirken Klangfarbenänderungen beim Rückwurf. Nach Blauert und Braasch [14, S. 107] kann beim Auftreten von Reflexionen das Hörereignis auf drei mögliche Arten von den Orten von Primär- und Sekundärschall abhängen. Im ersten Fall fließen die Ortsinformationen beider Schallereignisse in das Hörereignis ein: es kommt zu einer Phantomschallquelle. Im zweiten Fall kann der Präzedenzeffekt für das Hörereignis ausschlaggebend und nur der Direktschall richtungsbestimmend sein. Drittens kann es zu zwei Hörereignissen kommen: diese hängen von den Orten des Direktschalls und des Echos ab [14]. Im Schnittpunkt von Frontal- und Horizontalebene spannt sich beidseitig jeweils ein kegelförmiger Bereich auf, der sog. *Kegel unsicherer Lokalisation* (Cone of Confusion, COC), innerhalb dessen es Konstellationen

[1] Unterhalb 800 Hz ist unter Annahme $a = 21,5\ cm$ und $\vartheta = 20$ °C mit einer resultierenden ITD von 626 μs die ILD sehr gering

gleicher ITDs und ILDs gibt [12, S. 676, Abb. 11.34]. Der Mensch mit seinen starren Außenohren hilft sich mit minimalen Kopfbewegungen aus diesem Dilemma. Diese willkürlichen und unwillkürlichen Peilbewegungen unterstützen nicht nur die schwierige Lokalisation aus Schalleinfallsrichtungen innerhalb des COC durch Generierung aussagekräftiger ITDs und ILDs, sondern auch beim Auftreten der sog. *Vorne-Hinten Ambiguität* (Front-Back Ambiguity, FBA). Dieser Begriff steht für die unsichere Lokalisation in der Vorn-Hinten-Richtung und kann bei der Verwendung von nicht-individuellen HRTFs auftreten, so z. B. bei Kunstkopfaufnahmen. Die minimalen Kopfbewegungen stellen auch hier zusätzliche binaurale Cues zur Verfügung. Untersuchungen [19, 20, 21] belegen deren wichtigen Beitrag zum Erfolg der Lokalisation [22, S. 21]. Die Trägheit dieser Kopfbewegungen begrenzt allerdings deren Anwendbarkeit auf Signaldauern mit $t \geq 200$ ms. Darüber hinaus kann auch eine hohe Lautheit innerhalb richtungsbestimmender Bänder für eine FBA ursächlich sein [14]. Der oft verwendete Begriff der Vorne-Hinten-Vertauschung ist insofern irreführend, als dass es meist nicht zu einer Vorne-Lokalisation einer rückwärtigen Schallquelle kommt, wohl aber umgekehrt. Diesbezüglich stellen Xiang et al. [23] eine Koinzidenz zwischen frontaler Lokalisationsschwäche und dem Sehfeld fest. Eine quantitative Untersuchung führte Seeber [15, S. 83, 112] durch und konnte damit die Bedeutung der visuellen Fixation für die Lokalisation belegen. In seinen Versuchen sank die Zahl der FBAs um etwa die Hälfte, von $6,5\%$ auf $3,5\%$ wenn zusätzlich zum akustischen auch ein visueller Reiz in Form eines Lichtpunktes dargeboten wurde [ebda.]. Bei der Verarbeitung verschiedener Ohrsignalmerkmale können sich diese bei gegensätzlicher Richtungswirkung gegenseitig kompensieren. Dieses sog. *Trading* kann eine Aufsplittung des Hörereignisses bewirken, so dass die jeweilig charakteristischen Anteile aus verschiedenen Richtungen kommend lokalisiert werden. [14, S. 98]

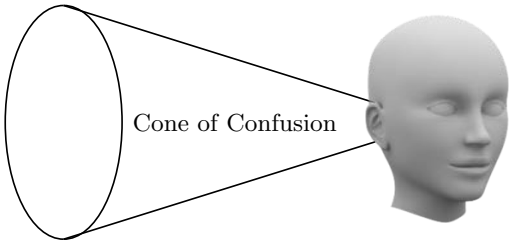


Abb. 2: Cone of Confusion

2.2.2 Entfernungsabschätzung

Im Gegensatz zu den Mechanismen des Richtungshörens sind die der Entfernungsabschätzung weniger komplex, insgesamt relativ schlecht ausgeprägt und beruhen mehr noch auf Lernprozessen. [14] Dabei werden das Verhältnis von Primär- und Sekundärschall, Klangfarbe bei breitbandigen Signalen, ITDGs, Bewegung und Lautheit ausgewertet und mit vergangenen Hörerfahrungen verglichen. Während in Räumen auch die ITDGs zur Entfernungsabschätzung herangezogen werden, ist es im Freifeld bis zum akustischen Horizont von ≈ 15 m hauptsächlich die frequenzunabhängige Pegeländerung nach $1/r$. Demzufolge erscheinen dort laute Hörereignisse näher als leise. [ebda.]

Interessant hinsichtlich einer Auralisation von Distanzen ist der von Seeber [15, S. 14] genannte Wert für die wahrgenommene Entfernungsverdopplung bei erst 20 dB anstelle der physikalisch messbaren 6 dB. Hörversuche von Zahorik [24] zeigen, dass die erwartete Entfernung durchgängig weit unter der tatsächlichen bleibt. Diese empirisch gefundene Beziehung zwischen realer Schall-

ereignisentfernung und wahrgenommener Hörereignisentfernung fasst Zahorik wie folgt in eine Gleichung[1]:

Gleichung 2.3: Zusammenhang von realer und erwarteter Schallereignisentfernung nach Zahorik [24]

$$r' = k \cdot r^a$$

$$\begin{aligned} r &= \text{reale Entfernung} \\ r' &= \text{erwartete Entfernung} \\ a &= 0,32 \\ k &= 1,00 \end{aligned}$$

Hinsichtlich der Audiosynthese in der App kann für den Fall, dass die physikalisch richtige Pegelabnahme keine plausiblen Ergebnisse liefert, auf diesen Ansatz zurückgegriffen werden. Grundsätzlich hängt das Entfernungshören auch von nicht-akustischen Parametern wie der Erfahrung mit der Schallquelle oder dem Vorhandensein visueller Cues ab. Außerdem ist zwischen geschätzter Entfernung und der tatsächlichen Entfernung zu differenzieren. Unterschieden werden nach Blauert und Braasch [14] drei Entfernungsbereiche. Im Nahbereich mit $r \leq 25$ cm kommt es zu starken Änderungen der monoauralen und binauralen Übertragungsfunktionen. So bewirkt eine Annäherung extreme ILDs und charakteristische Resonanzmuster der Ohrmuschel. Damit einher geht eine Zunahme des Gesamtschallpegels. Zusätzlich kommt es zu einer interauralen Parallaxe bzgl. der vom Gehör erkennbaren spektralen Änderung der Ohrsignale. Es kommt ähnlich wie in der Optik zu einem schnelleren Vorbeiziehen naher bewegter Quellen als bei Entfernteren. Bereits im mittleren Entfernungsbereich $r \approx 25$ cm bis 15 m ist die Kenntnis des Signals eine starke Einflussgröße für eine Abschätzung auf Grundlage der frequenzunabhängigen Schallpegeländerung und der pegelabhängigen Klangfarbe breitbandiger Signale. In kürzerer Distanz ist die ITDG länger, der Direktschallpegel ist im Verhältnis zum Diffusschallpegel höher und es treten frequenzabhängige ILDs auf. Schalleinfallsrichtung und Signaleigenschaften bestimmen den jeweiligen Einfluss dieser Merkmale auf die Lokalisation. Die Entfernungsabschätzung von großen Distanzen mit $r > 15$m, die über den akustischen Horizont hinausgehen, ist schwierig und wird wiederum durch Erfahrungen mit der Signalquelle und/oder gleichzeitige visuelle Informationen unterstützt. Mit wachsender Entfernung spielen ITDGs eine immer geringer werdende Rolle, stattdessen kommt es zu einer zunehmenden Höhenabsenkung durch die Dissipationsdämpfung der Luft und einem zum Direktschall verhältnismäßig höheren Diffusschallpegel [ebda.].

2.2.3 Im-Kopf-Lokalisation

Bei der Audiowiedergabe über Kopfhörer kommt es nicht zu den natürlich auftretenden Reflexionen an Kopf und Oberkörper. Die dadurch fehlenden, aber für die Lokalisation wichtigen Klangverfärbungen, gelten als eine Ursache der *Im-Kopf-Lokalisation* (Inside-the-Head Locatedness, IHL),

[1] Die Parameter a und k ermittelte er zur Funktionsanpassung aus den Messdaten.

da diese Klangverfärbungen zur binauralen Dekorrelation beitragen. Allgemein gilt ein hoher interauraler Korrelationsgrad als eine Hauptursache dieses psychoakustischen Effektes [14, S. 99], der man aber durch Anwendung geeigneter Entzerrfilter begegnen kann [ebda.]. Plenge [25] nennt darüber hinaus neben einer fehlerhaften Abbildung kopfbezogener Cues durch Mängel bei der äußerlichen Nachbildung des Kunstkopfes auch rein psychologische Parameter. So gibt es einen starken Zusammenhang zwischen der Stimmigkeit der akustischen und visuellen Informationen sowie der Propriozeption der Kopfbewegungen mit dem Auftreten der IHL im Falle von Inkonsistenzen. Sehr relevant für diese Arbeit ist die Erkenntnis der Bedeutung der Adaption an die akustische Szene und die der visuellen Reize. Da hier eine rein akustische Augmentierung angestrebt wird, ist einerseits mit Einschränkungen bei der Plausibilität der Auralisation aufgrund fehlender optischer Informationen zu rechnen und andererseits kommt der Genauigkeit des Echtzeit-Head-Trackings eine große Bedeutung zu. Als weitere Einschränkung könnte die Verwendung nicht-individueller HRTFs gesehen werden, die nach Untersuchungen Seebers [15, S. 60f.] ebenfalls eine IHL begünstigen kann, wenn die Geometrie des Dummy-Heads oder der Versuchsperson stark von der eigenen abweicht. Bei der Auralisation mittels nicht-individueller HRTFs kam es bei seinen Hörversuchen zu IHL bei voraus liegenden Richtungen. Offenbar stellen diese mit der zusätzlich dort möglichen FBA die größte Herausforderung in der Horizontalen dar. Auch Plenge [25, S. 252] stellt bei fehlenden ITD durch gleichphasige Signale bei seitlicher Lautsprecherwiedergabe Fälle von IHL fest. Bei der Verwendung individueller HRTFs kommt es bei Seeber [15] hingegen trotz fehlender Kopfhörerentzerrung ausschließlich zu *Außer-Kopf-Lokalisationen (Outside-the-Head Localizations)* (OHLs). Nach Plenge [25, S. 251] kommt als mögliche Ursache von IHL auch eine falsche Wiedergabelautstärke kopfbezogener Audiosignale in Betracht. Als möglicher Grund kommt die Pegelabhängigkeit der Isophone in Frage, durch welche die spektralen binauralen Cues verzerrt werden können.

2.2.4 Physiologische Auswertung binauraler Ohrsignale

Nach Trahiotis et al. [26, S. 238 ff.] ist bei der Auswertung der Ohrsignale der Grad ihrer Ähnlichkeit von großer Bedeutung für die Richtungsbestimmung. Das 1948 von Jeffress [27] postulierte Modell einer *Binauralen Verzögerungskette* besagt, dass für eine bestimmte Schalleinfallsrichtung die ITDs durch neuronale Laufzeitunterschiede kompensiert und diese Schalleinfallsrichtung durch eine erhöhte Feuerungsrate sog. *Koinzidenzneuronen* detektiert wird. Nach Dau [28] wurden 1990 solche Neuronen in der Medialen Oberen Olive im Hirnstamm nachgewiesen. Mathematisch vereinfacht kann dieser biologische Prozess durch die *Kreuzkorrelationsfunktion* ausgedrückt werden.

Gleichung 2.4: Normierte Kreuzkorrelation nach Blauert und Braasch [14, S.105]

$$R_{norm,x,y}(\tau) = \frac{\lim\limits_{T\to\infty} \frac{1}{2T} \int\limits_{-T}^{+T} x(t) \cdot y(t+\tau)\,\mathrm{d}t}{x(t)_{eff} \cdot y(t)_{eff}}$$

$x(t) :=$ rechtes Ohrsignal,
$y(t) :=$ linkes Ohrsignal

Der *interaurale Kohärenzgrad* definiert den Grad der Ähnlichkeit beider Ohrensignale:

Gleichung 2.5: interauraler Kohärenzgrad nach Blauert und Braasch [14, S.106]

$$k = \underset{\tau}{max} |\, R_{norm,x,y}(\tau)| \text{ , für } -1 \text{ ms} \leq \tau \leq +1 \text{ ms}$$

Dabei bedeutet $\tau = 1$ völlige Ähnlichkeit beider Signale und analog $\tau = 0$ völlige Unähnlichkeit, wobei frequenzunabhängige Pegel- und Laufzeitunterschiede nicht zur Inkohärenz beitragen. Bereits ein Korrelationsgrad $\tau \geqq 0.66$ geht mit zunehmender IHL einher – ein psychoakustischer Effekt, bei dem die Schallquelle entlang der interauralen Achse zwischen den Ohren lokalisiert wird und der z. B. durch monophone Signalwiedergabe über Kopfhörer herbeigeführt werden kann. Signaltechnisch können die interauralen Signalunterschiede auch aus der Definition der HRTF abgeleitet werden [15, S. 37].

Gleichung 2.6: interaurale Signalunterschiede

$$H_\Delta(\varphi, \theta, \omega, r) = \frac{H_r(\varphi, \theta, \omega, r)}{H_l(\varphi, \theta, \omega, r)}$$

Gleichung 2.7: interaurale Pegeldifferenz

$$L_\Delta(\varphi, \theta, \omega, r) = 20 \cdot log_{10}(|H_\Delta(\varphi, \theta, \omega, r)|)$$

Gleichung 2.8: interaurale Phasendifferenz

$$\varphi_\Delta(\varphi, \theta, \omega, r) = arg\{H_\Delta(\varphi, \theta, \omega, r)\}$$

Gleichung 2.9: interaurale Gruppenlaufzeit

$$\begin{aligned}\tau_\Delta(\varphi,\theta,\omega,r) &= \frac{\mathrm{d}\varphi_\Delta(\varphi,\theta,\omega,r)}{\mathrm{d}\omega} \\ &= \frac{\mathrm{d}}{\mathrm{d}\omega}\, arg\{H_\Delta(\varphi,\theta,\omega,r)\end{aligned}$$

Mit abnehmender *Interaurale Kohärenz* (IK) kommt es zu einer räumlichen Verbreiterung des Hörereignisses, bis bei zu großer Unähnlichkeit beider Ohrsignale getrennte Einzelhörereignisse auftreten. Durch das natürliche Übersprechen der Ohrsignalkanäle kann im Freifeld der Fall $\tau = 0$ nicht eintreten. [14]

2.3 Akustische Einflüsse von Umweltfaktoren

Die Schallausbreitung im Freien findet meist in einem teilbegrenzten Halbfreifeld statt und wird von der Art der geometrischen Wellenausbreitung, der Luftabsorption und der Struktur des Schallausbreitungsgebietes beeinflusst. Die Art der geometrischen Ausbreitung hängt vom Wellentyp ab. In dieser Arbeit werden keine Linienquellen oder Flächenquellen auralisiert, daher beschränkt sich die Betrachtung auf Punktquellen mit kugelförmiger Ausbreitung. Diese kann wegen des Falles eines einzigen Messpunktes in Form des Hörers als ebene Welle approximiert werden, wobei weiterhin mit einer Pegelabnahme mit $1/r$ gerechnet wird. Der Einfluss der Entfernung zur Schallquelle teilt sich in einen frequenzunabhängigen Anteil und den frequenzabhängigen Anteil der Dissipationsdämpfung. Die Struktur des Terrains schließt Bodeneffekte, Reflexion an Flächen und Abschirmung durch Hindernisse mit ein. Ausführliche Beschreibungen der Zusammenhänge und deren mathematischen Berechnung findet sich u. a. in Rossing et al. [29, S. 113-143] und der ISO 9613-1/2 [2, 3]. Untersuchungsergebnisse speziell zum Einfluss von Temperatur und Luftfeuchtigkeit auf die Schallausbreitung sind auffindbar bei Harris et al. [30]. Die Gesamtdämpfung im Freien kann in die folgenden Teildämpfungen zerlegt werden:

Gleichung 2.10: Schalldämpfung im realen teilbegrenzten Halbfreifeld

$$A = A_{geom} + A_{diss} + A_{bod} + A_{bar} + A_{div}$$

$$\begin{aligned}A_{geom} &= \text{Dämpfung durch die geometrische Ausbreitung} \\ A_{diss} &= \text{Dämpfung durch die Dissipation der Luft} \\ A_{bod} &= \text{Dämpfung durch Absoptionsgrad des Bodens} \\ A_{bar} &= \text{Dämpfung durch Barrieren im Terrain} \\ A_{veg} &= \text{Dämpfung durch Vegetation im Terrain} \\ A_{div} &= \text{Dämpfung durch diverse weitere Einflussgrößen}\end{aligned}$$

Unter A_{div} sind Teildämpfungen zusammengefasst, die aufgrund fehlender Relevanz für die App in der Implementierung nicht berücksichtigt werden. Dazu gehören die meteorologische Korrektur, mit den Eigenschaften der Schallausbreitung im turbulenten Medium Luft, sowie der Fall einer Inversionswetterlage.

2.3.1 Meteorologische Cues

Bei der Schallausbreitung im Außenbereich können meteorologische Veränderungen innerhalb kürzester Zeit Pegelschwankungen von $10 - 20$ dB in Abhängigkeit vom Schallausbreitungsweg verursachen. [110] Im Folgenden werden mit der Temperatur und relativen Feuchtigkeit zwei physikalische Eigenschaften des Mediums Luft herausgegriffen und deren Einfluss auf die Schallübertragung beleuchtet.

Einfluss der Lufttemperatur

Die Schallgeschwindigkeit ist stark von der Temperatur abhängig. Die Schallgeschwindigkeit wiederum beeinflusst mit $f = c/\lambda$ die Frequenz und damit die wahrgenommene Tonhöhe.

Gleichung 2.11: Schallgeschwindigkeit in Abhängigkeit von der Temperatur

$$c = c_0 \cdot \sqrt{\left(1 + \left(\frac{\vartheta}{273,15}\right)\right)}, \qquad \left[\frac{m}{s}\right]$$

$c =$ resultierende Schallgeschwindigkeit
$c_0 = 331,5$ m/s, Referenzschallgeschwindigkeit bei $\vartheta = 0$°C
in trockener Luft

vereinfacht gilt auch:

Gleichung 2.12: Schallgeschwindigkeit in Abhängigkeit von der Temperatur (vereinfacht)

$$c \approx 331,5 + (0,6 \cdot \vartheta) \Rightarrow \Delta\vartheta = 1°\text{C} \rightarrow \Delta c = 0,6 \left[\frac{m}{s}\right]$$

Damit ergibt sich für gewöhnliche mitteleuropäische Temperaturverhältnisse zwischen -20°C und $+40$°C eine maximal mögliche Schallgeschwindigkeitsschwankung von $\Delta c = 36$ m/s. Die Frequenzänderung Δf beträgt $\approx 0,75$ Hz für $\Delta\vartheta = \pm 1$°C, was Auswirkungen auf die Tonhöhenempfindung besonders im tieffrequenten Bereich hat. Ein 100 Hz Sinuston bei $\vartheta = -20$°C hätte demnach bei $+40$°C eine Frequenz von ≈ 145 Hz – das entspricht mehr als 6 Halbtönen. Obwohl dieser Effekt aufgrund des logarithmischen Charakters der Tonhöhenwahrnehmung mit der

Frequenz abnimmt, unterschreitet er erst ab $6,429$ kHz für das genannte $\Delta\vartheta$ die JND der Tonhöhenwahrnehmung. Als Schlussfolgerung werden für die Simulation räumlicher Cues jahreszeitlich differenzierte OIRs verwendet.

Einfluss der Luftfeuchtigkeit

Als Dissipationsdämpfung bezeichnet man die mit der Frequenz zunehmende Luftschalldämpfung aufgrund von Reibungs- und molekularen Relaxationsverlusten. Durch diese Prozesse wird Schallenergie in thermische Energie umgewandelt. Für hohe Frequenzen und lange Distanzen wirkt das Medium Luft als ein Tiefpassfilter, dessen Charakteristik von der relativen Luftfeuchtigkeit abhängig ist.
Die relative Luftfeuchtigkeit unterliegt täglichen Schwankungen, welche im Sommer am stärksten sind, und erreicht meist am Nachmittag ihren Spitzenwert [29, S.116]. Für die Betrachtung der entfernungsabhängigen Schallpegeländerung im Freien muss der Einfluss der Dissipationsdämpfung A_{diss} zu der frequenzunabhängigen Abnahme des Schalldruckpegels durch A_{geom} addiert werden. Bei einer vereinfachenden Betrachtung einer Schallquelle mit omnidirektionaler Ausbreitung wird der Richtwirkungsfaktor $DI = 0$ dB gesetzt. Befindet sich der Abstrahlort direkt auf einer schallharten Unterlage gilt $DI = 11 - 10 \cdot log_{10}(2\pi) \approx 3$ dB. [110]

Gleichung 2.13: Schallausbreitung im Freifeld

$$L_p = \underbrace{L_w - 20\, log_{10} r - A_{geom} + DI}_{\text{ideales Freifeld ohne Reflexionen}} - A_{diss} \qquad [dB]$$

$$A_{geom} = \left[20\, log_{10}\left(\frac{r_2}{r_1}\right) + 10\, log_{10}\left(\frac{W_{ref}\rho c}{P_{ref}4\pi}\right)\right], \qquad [dB]$$

$$A_{diss} = \alpha \cdot r, \text{ Dämpfung im Abstand } r \text{ m} \qquad [dB]$$

$$r = \text{ Abstand von der Schallquelle} \qquad [m]$$

$$\rho c = 400 \text{ mks } rayls, \text{ Schallkennimpedanz} \qquad \left[\frac{kg}{m^2 s}\right]$$

$$DI = 10\, log_{10}\, Q, \text{ Richtwirkungsindex} \qquad [dB]$$

$$Q = \frac{I_\theta}{I_{mean}} = \frac{P_\theta^2}{P_s^2}, \text{ Richtwirkungsfaktor}$$

$$L_p = L_w - 20\, log_{10}\, r - 11 + DI - A_{diss} \qquad [dB]$$

Die Absorptionsgrade α_{Luft} sind der Tabelle in ISO 9613-1 [2, S. 5] für Bandmittenfrequenzen bis 8 kHz entnehmbar. Für Standarddruckverhältnisse in einem Temperaturbereich von $\vartheta = -20$°C bis $+50$°C kann α in Abhängigkeit von Frequenz, Temperatur und Luftfeuchtigkeit wie folgt berechnet werden.

Gleichung 2.14: Absorptionsgrad der Luft [ebda.]

$$\alpha_{Luft} = 8,686 f^2 \left\{ \left[1,84 \cdot 10^{-11} \left(\frac{p_a}{p_r}\right)^{-1} \left(\frac{T}{T_0}\right)^{1/2} \right] + \left(\frac{T}{T_0}\right)^{-5/2} \left[0.01275 \frac{e^{-2239,1/T}}{F_{r,O} + f^2/F_{r,O}} + 0.1068 \frac{e^{-3352/T}}{F_{r,N} + f^2/F_{r,N}} \right] \right\}$$

$$F_{r,N} = \frac{p_a}{p_r} \left(\frac{T}{T_0}\right)^{-1/2} \left(9 + 280H \cdot e^{\left\{-4,17\left[\left(\frac{T}{T_0}\right)^{-1/3} - 1\right]\right\}} \right)$$

$$F_{r,O} = \frac{p_a}{p_r} \left(24 + 4,04 \cdot 10^4 H \frac{0.02 + H}{0,391 + H} \right)$$

$F_{r,N} =$ Relaxationsfrequenz Stickstoff (Hz)

$F_{r,O} =$ Relaxationsfrequenz Sauerstoff (Hz)

$H_r =$ relative Luftfeuchtigkeit $(\%)$

$h = 34$, Höhe über dem Meeresspiegel in Berlin (m)

$p_a = p_r \times e^{-\frac{h}{h_s}} = 100,924$, Atmosphärendruck (Berlin) (kPa)

$h_s = \dfrac{R_{fL} \cdot T}{g} = const.$ (m)

$R_{fL} = \dfrac{R_{tL}}{1 - \left(\frac{H \cdot e_{sat}}{p_a}\right) \cdot \left(1 - \frac{R_{tL}}{R_D}\right)}$, Gaskonstante feuchte Luft $(J/kg \cdot K)$

$R_{tL} = 287,058$, Gaskonstante trockene Luft $(J/kg \cdot K)$

$R_D = 461,51$ Gaskonstante Wasserdampf $(J/kg \cdot K)$

$E_W(t) = 611,2$ hPa $e^{\frac{17,62t}{243,12°C+t}}$, Sättigungsdruck Wasser (-45 $\leq t \leq$ +60°C) (hPa)

$g = 9,80665$, Fallbeschleunigung (m/s^2)

$p_r = 101,325$ relativer Atmosphärendruck (kPa)

$t =$ Temperatur (C)

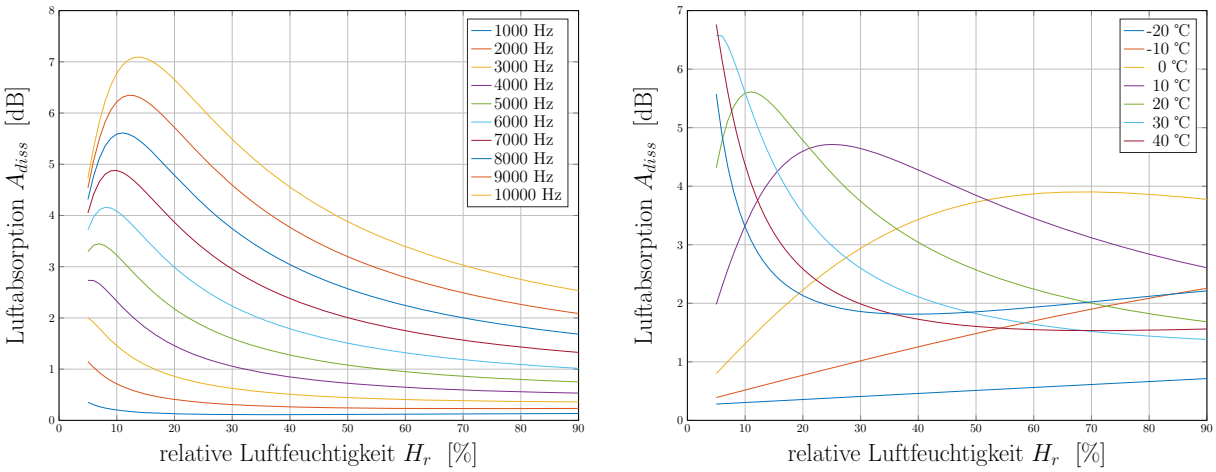
$T =$ Temperatur (K)

$T_0 = 293,15$ Referenztemperatur (K)

Die Luft ist zu $78,08\%$ aus Stickstoff und zu $20,95\%$ aus Sauerstoff zusammengesetzt, daher ist für die Berechnung von α_{Luft} die Einbeziehung der Eigenschaften dieser beiden Gase hinreichend. Die Schwingungsrelaxation hängt von der Relaxationsfrequenz und von der relativen Luftfeuchte des Gases ab, der Absorptionsgrad zusätzlich von der Lufttemperatur.

Der folgende MATLAB-Plot in Abbildung 3a zeigt – abgesehen vom Fall sehr trockener Luft – eine mit zunehmender relativen Luftfeuchtigkeit abnehmende Luftabsorption für $\vartheta = 20°\text{C}$. Abbildung 3b zeigt zudem den Einfluss der Lufttemperatur nach Formel 2.14.

Abb. 3: Dissipationsdämpfung der Luft für reine Sinustöne bei $L_p = 100$ dB in einem Abstand von 25 m



(a) Dissipationsdämpfung für $\vartheta = 20$ °C $= const.$

(b) Dissipationsdämpfung $f = 8$ kHz $= const.$

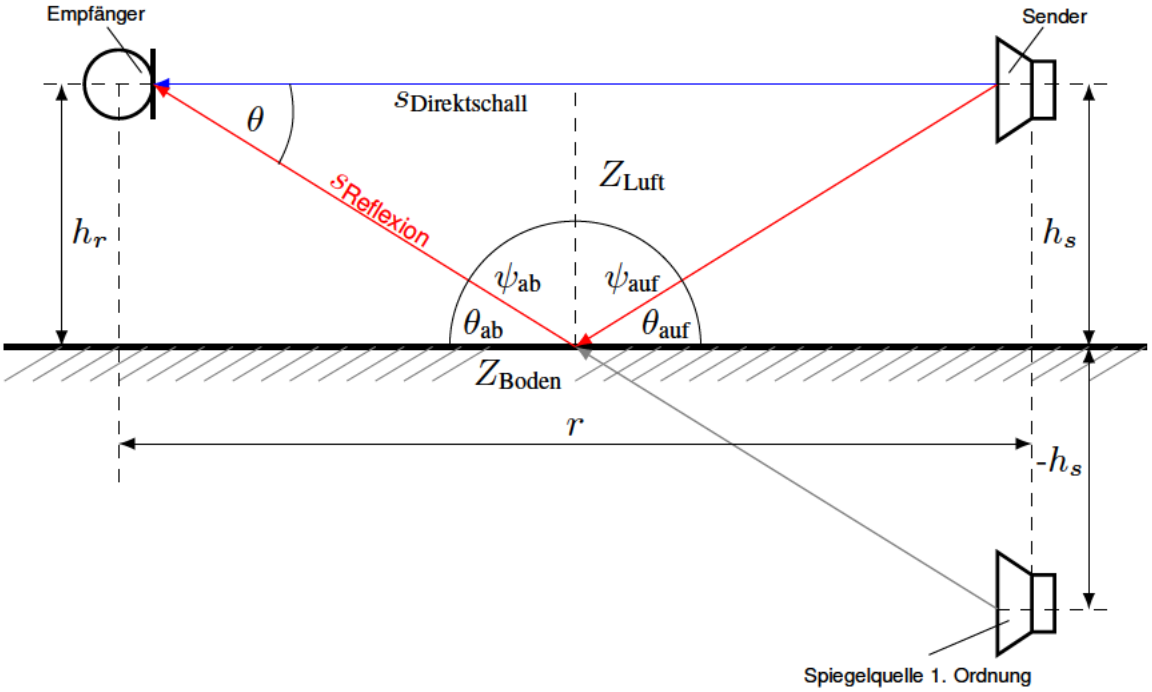
Die Dissipationsdämpfung leistet einen wesentlichen Beitrag zur entfernungsabhängigen Klangfarbenveränderung, daher wird zur plausibleren Simulation in Betracht gezogen, den Dissipationsfilter mit maximalen Dämpfungsparametern zu speisen, wie sie bei geringerer bis mittlerer relativen Luftfeuchtigkeit zu erwarten wären.

2.3.2 Einflüsse von Boden und Terrain

Da in Außenumgebungen unter freiem Himmel die Bodenreflexion nicht an ihrer vertikalen Ausbreitung gehindert[1] und für den Nahbereich der Schallquelle Barrierefreiheit angenommen wird, kann für die Implementierung ein vereinfachendes Spiegelschallquellenmodell 1. Ordnung herangezogen werden, bei dem die Punktschallquelle mit sphärischer Ausbreitung im Moment des Auftreffens auf den Boden als ebene Welle betrachtet wird [9, S. 41].

[1] Eine Ausnahme bildet der in der Implementierung nicht berücksichtigte Fall der Inversionswetterlage, bei der es wegen $\Delta\vartheta$ zwischen einer kalten Luftschicht am Boden und einer wärmeren Luftschicht darüber zu einem Impedanzsprung aufgrund unterschiedlicher Schallgeschwindigkeiten beider Schichten und damit zu einer Teilreflexion zurück zum Boden hin kommt. [31]

Abb. 4: Spiegelschallquellenmodell 1.Ordnung



Auf der Schallübertragungsstrecke kommt es zu Überlagerungen und Auslöschungen zwischen Direktschall und Bodenreflexion. Bei Reflexionen an schallharten Untergründen können durch deren Laufzeitunterschiede und der geringen Pegeldifferenz von Direktschall und Reflexion hörbare Kammfiltereffekte auftreten. An den Stellen ungeradzahliger Vielfache der ersten Notchfrequenz f_{n_0} kommt es zu subtraktiven, an den geradzahligen Vielfachen zu negativen Interferenzen.
Brunner [32] stellt in seiner Magisterarbeit über die *Perzeptive Bewertung von kammfilterartigen linearen Verzerrungen in der Audiotechnik* eine deutlichere Wahrnehmbarkeit von Peaks gegenüber den Dips fest. Ungedämpfte Überlagerungen können demnach zu einem um 6 dB höheren Pegel an den Stellen f_{n_m} führen. Nach Müller und Möser [33, S. 199] kommt es zusätzlich bei höheren Frequenzen und sehr flachem Schalleinfall zu einer durch die Phasendrehung der Reflexion verursachten destruktiven Interferenz in bestimmten Frequenzbereichen. Auch sind bei geringen Δs zwischen Direktschall und Reflexion diese als kohärent anzusehen, was für sehr tiefe Frequenzen eine phasenrichtige Addition mit einem 3 dB höheren Schalldruckpegel im Vergleich zur inkohärenter Überlagerung nach sich zieht [ebda.]. Im realen Schallfeld hingegen konnte Theile [34] in Hörversuchen eine Diskrepanz zwischen wahrgenommener Klangfarbe und physikalischen Kammfiltereffekten bei binauraler Darbietung nachweisen. Da diese Beobachtung bei monoauralem Hören nicht gemacht werden konnte, schlussfolgerte er auf eine kognitive Ausblendfunktion des auditiven Systems und integrierte dies in sein Assoziationsmodell. Daraus ist die höhere Empfindlichkeit für signaltechnisch herbeigeführte kammfilterartige lineare Verzerrungen mit der

fehlenden auditiven Erfahrung des Hörers erklärbar. Schlussfolgernd kann nicht ausgeschlossen werden, dass bei einer signaltechnischen Überlagerung von Direktschall und simulierter Reflexion bei schneller Fortbewegung des Hörers Flanging- und Phasingeffekte auftreten können.
Nach Müller und Möser [33, S. 199] gilt für größere Abstände von der Schallquelle für die reflektierte Welle eine Zusatzdämpfung von 3 dB zu der geometrische Ausbreitungsdämpfung nach dem $1/r$-Gesetz aufgrund von Streuungs- und Absorptionseffekten durch die Bodenbeschaffenheit.
Der Bodeneffekt beinhaltet Streuungseffekte in Abhängigkeit von der Rauhigkeit der Oberflächenstruktur, Absorptionsdämpfung von nicht schallhartem Untergrund und destruktive und konstruktive Interferenzen, hervorgerufen durch die Überlagerung von phasenverschobenem Direktschall und Bodenreflexion [29, S. 120]. Diese Phasenverschiebungen begründen sich durch unterschiedlich lange Signallaufwege und dem von der Bodenimpedanz abhängigen Phasensprung der Bodenreflexion [35, S. 33]. Der Reflexionsfaktor ist abhängig von der Frequenz, der Bodenimpedanz und dem Einfallswinkel. Farnell [36, S. 71] stellt einen besonders starken Einfluss der Bodendämpfung auf das Signalspektrum der reflektierten Welle in einem Abstand unterhalb von 4 m zur Schallquelle fest, was mit einem abnehmenden Absorptionsgrad bei größer werdendem Einfallswinkel ψ begründbar ist. Zhou und Zha [37] konnten mit einer eigenen Messmethode ebenfalls einen kontinuierlichen Abfall des Absorptionsgrades eines porösen Absorbers mit kleiner werdenden Einfallswinkel beobachten. Mathematisch ist dies in Formel 2.16 nachvollziehbar: Für sehr flache Einfallswinkel $\sin\psi \to 0$ nähert sich R_p dem Wert -1, was einer Totalreflexion mit Phasensprung von $180°$ entspricht.
Die Richtung der Bodenreflexion ergibt sich bei kopfbezogener Betrachtung nach dem Huygens-Prinzip als geometrische Spiegelung mit Einfallswinkel ψ_{auf} gleich Abstrahlwinkel ψ_{ab} bei Reflexion an einer ebenen Fläche A_r für alle Frequenzen $f \ll A_r$. Ihr jeweiliger geometrischer Wellencharakter bleibt dabei erhalten. Bei unebenen Flächen kommt es darüber hinaus für Frequenzen, deren λ in die Größenordung der Unebenheiten fällt, zu diffuser Abstrahlung.[38, S. 285]
Die Amplituden- und Phasenänderung der Reflexion wird bei lokal reagierendem Medium von der komplexen Bodenimpedanz $Z_{\text{Boden}} = R_{\text{Boden}} + jX_{\text{Boden}}$ bestimmt. Die Stärke ihres Einflusses hängt auch mit der Senderhöhe h_s oder Empfängerhöhe h_r zusammen. Unter niedrigen Schalleinfallswinkeln kommt es zur Ausbildung von Schallschattenbereichen hervorgerufen von niedrigen Frequenzen der Bodenwelle [39, S. 114] – im umgekehrten Fall bei großen h_r und/oder h_s ist ihr Einfluss auf das Signal am Empfangsort eher gering. Beide Fälle befinden sich jedoch außerhalb des Anwendungsbereiches der geplanten Simulation. Für diese ergibt sich nun das Problem, dass einerseits die Art der Bodenoberfläche mittels Ferndiagnose über ein Display nur grob abschätzbar ist und daher nur im Rahmen prototypischer Presets berücksichtigt werden kann. Zum anderen sind im Halbfreifeld in erster Linie der Direktschall und dessen Überlagerung mit der Bodenreflexion akustisch wirksam, was die Einbeziehung des winkelabhängigen Reflexionsfaktors nahelegt, da die im Diffusfeld ermittelten Absorptionsgrade einen über alle Richtungen gemittelten Durchschnittswert widerspiegeln.

Gleichung 2.15: Diffusfeld-Reflexionsfaktor

$$r_{diff} = \sqrt{1 - \alpha_{diff}}$$

$$\begin{aligned} \alpha_{diff} &= \text{Diffusfeld-Absorptionsgrad} \\ r_{diff} &= \text{Diffusfeld-Reflexionsfaktor} \end{aligned}$$

Darüber hinaus beeinhaltet r_{diff} die Eigenschaften des äußeren Raumschallfeldes [40], was im Widerspruch zum hier zu simulierenden Fall eines nur teilbegrenzten Halbfreifeldes stünde. Die Verwendung des komplexen Reflexionsfaktors setzt jedoch die Kenntnis der komplexen Bodenimpedanz voraus, welche meist nur mit relativ hohem Aufwand in-situ gemessen werden kann, da selbst Laborproben nicht repräsentativ für natürliche inhomogen geschichtete Böden sind [41, S. 499]. Eine Lösung für dieses Problem lässt sich bei Delany und Bazley [42] finden, deren empirisches Modell lediglich den Strömungswiderstand σ zur Berechnung der komplexen Bodenimpedanz als Inputparameter erfordert, welcher in gröberen Abstufungen als empirischer Wert vorliegt [43, 44]. Damit ist es möglich für verschiedene Bodeneigenschaften prototypische komplexe Reflexionsfaktoren zu ermitteln. Die darin enthaltende Reflexionsphase wurde von Stumpner und Goossens [45] in vergleichenden Simulationen mit und ohne deren Berücksichtigung als nicht wahrnehmbar eingeschätzt und wird daher im Folgenden nicht berücksichtigt. Lässt man Streuungseffekte an der Oberfläche unberücksichtigt, berechnet sich nach Piercy [39, S. 102] die winkelabhängige Dämpfung wie folgt.

Gleichung 2.16: winkelabhängiger Reflexionsfaktor

$$R = \frac{\sin\theta - \frac{Z_0}{Z_1}}{\sin\theta + \frac{Z_0}{Z_1}}$$

$$\begin{aligned} \theta &= \text{Schalleinfallswinkel, vgl. Abb. 4} \\ Z_0 &= \rho_0 \cdot c_0 = 415.1\text{; Schallkennimpedanz von Luft bei } \vartheta = 20°\text{C} && [Pa \cdot s/m] \\ Z_1 &= R_1 + jX_1\text{; Bodenimpedanz (lokal reagierende Oberfläche)} && [N \cdot s/m^3] \end{aligned}$$

Experimentell wurden folgende Werte ermittelt:

Gleichung 2.17: Bodenimpedanz nach Delany und Bazley [42, S. 107] – überarbeitet durch Miki [46, S. 21]

$$Z_1 = \rho_0 c_0 \left[1 + 5,50 \left(10^3 \frac{f}{\sigma} \right)^{-0.632} - j8,43 \left(10^3 \frac{f}{\sigma} \right)^{-0.632} \right]$$

mit $\sigma =$ längenbezogener Strömungswiderstand in Ausbreitungsrichtung

Tabelle 1: Strömungswiderstand einiger Bodenoberflächen nach Embleton [43] und Attenborough et al. [44, S. 141 ff.]

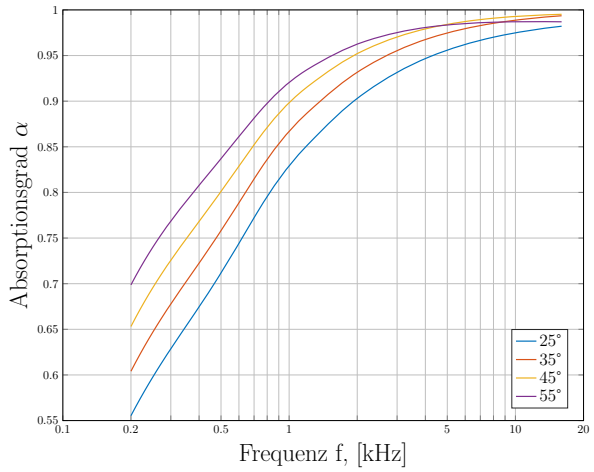
Bodenoberflächentyp	**Strömungswiderstand** σ $[kN\ s/m^4]$
trockener Neuschnee - älterer Zuckerschnee	5 - 50
Nadelwaldboden	20 - 80
lehmiger Sandboden, Grass, durchwurzelt	62 - 166
trockener Sand	134
Grassboden/Weide	150 - 300
feuchter Sand	479
feiner, fester Sandboden	800 - 2500
Schotter 12 - 25 mm Korngröße	1500 - 4000
verdichteter Gesteinsstaub	5000 - 20000
versiegelter Asphalt	$\approx$ 30000

Mit den nun vorhandenen Größen errechnet sich der Absorptionsgrad mit

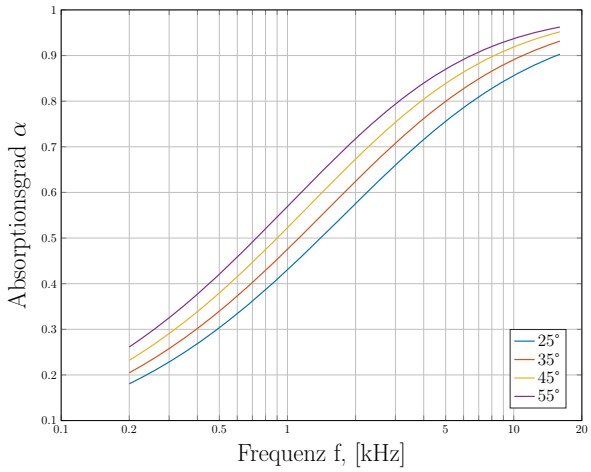
Gleichung 2.18: winkelabhängiger Absorptionsgrad

$$\alpha = 1 - |R|^2.$$

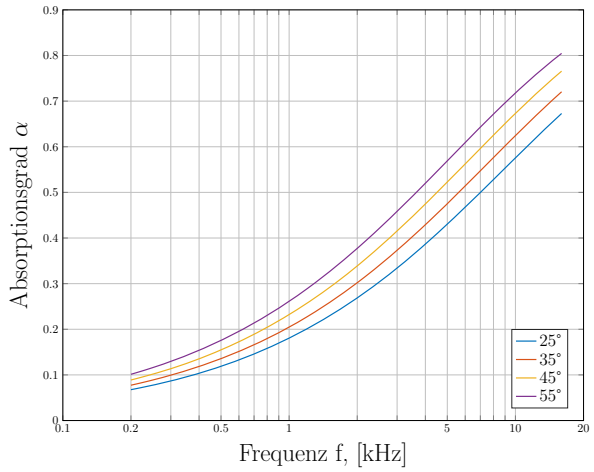
Abb. 5: winkelabhängiger Absorptionsgrad α nach Miki [46] (vgl. Formel 2.17) für ausgewählte Bodenoberflächen (vgl. Tab. 1)



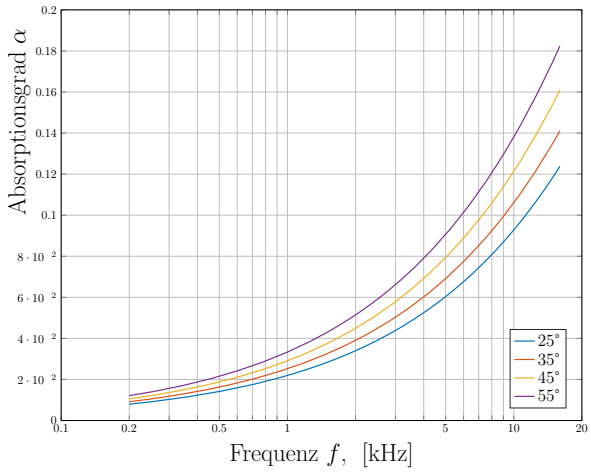
(a) Absorptionsgrade für Schnee: $\sigma = 200\ kN\ s/m^4$ und $\delta = 20\ cm$



(b) Absorptionsgrade für Wiese: $\sigma = 20\ kN\ s/m^4$ und $\delta = 15\ cm$



(c) Absorptionsgrade für Sand: $\sigma = 1000\ kN\ s/m^4$ und $\delta = 50\ cm$



(d) Absorptionsgrade für Asphalt: $\sigma = 30000\ kN\ s/m^4$ und $\delta = 10\ cm$

3 Stand der Technik: Mobile Dynamische Binauralsynthese

Das folgende Kapitel ist in drei Abschnitte untergliedert: im ersten werden grundlegende mathematische Operationen der Binauralsynthese vorgestellt, im zweiten deren technische Umsetzungen erläutert und im dritten erfolgt eine Einordnung des Projektes in den Kontext *Augmentierte Realität* (Augmented Reality, AR).

3.1 Systemtheoretische Grundlagen

Es folgt eine kurze Erläuterung der beiden wichtigsten signaltechnischen Methoden, die in dieser App zur Umsetzung der DBS angewendet wurden: die Schnelle Fourier-Transformation und die Schnelle Faltung.

3.1.1 Schnelle Fourier-Transformation

Die *Fouriertransformation* (Fourier Transform, FT) lässt sich direkt aus der Fourierreihe ableiten und basiert auf Fourier's Erkenntnis, dass sich periodische Funktionen durch eine Kombination harmonischer Schwingungen[1] darstellen lassen. Laplace kritisierte, dass die FT für ideale Signalformen wie z. B. die Rechteckfunktion nur eine Approximation leisten kann. Tatsächlich treten an den Sprungstellen sog. Überschwinger[2] aufgrund des Gibbs'schen Phänomens [47, S.118] auf, da reale Systeme aufgrund ihrer Tiefpasseigenschaft nicht die für Sprungstellen erforderlichen unendlich hohen Frequenzen übertragen können – die Vorteile der FT überwiegen jedoch.
Mit dem Gleichungspaar der FT ist es möglich, ein Signal vom Zeitbereich in den Frequenzbereich zu überführen und vice versa.

Gleichung 3.1: Fourier-Transformationspaar für kontinuierliche Signale [47, S. 90]

$$\text{Transformation:} \qquad X(f) = F\{x(t)\} = \int_{-\infty}^{\infty} x(t) e^{-j2\pi ft}\,\mathrm{d}t$$

$$\text{Rücktransformation:} \qquad x(t) = F^{-1}\{X(f)\} = \frac{1}{2\pi}\int_{-\infty}^{\infty} X(f) e^{j2\pi ft}\,\mathrm{d}f$$

Da Rechnersysteme mit endlichen, diskreten Werten arbeiten, ist eine computerfreundliche Näherung an die Fourierintegrale nötig. Das Konzept dazu liefert die *Diskrete Fouriertransformation* (Diskrete Fourier Transform, DFT) als approximatives Pendant zur FT – ein performanter Algorithmus zu deren effizienter Umsetzung auf digitalen Systemen ist die *Schnelle Fourier Transformation* (Fast Fourier Transform, FFT). Auch für die DFT gilt die Linearität, d. h. dass die Linearkombination von Signalen gleich der Linearkombination ihrer DFTs ist und somit bilden die Transformationspaare $x_1[n]+X_1[k]$ und $x_2[n]+X_2[k]$ mit $k_1x_1[n]+k_2x_2[n]$ und $k_1X_1[k]+k_2X_2[k]$

[1] Die Darstellung kann erfolgen durch Addition von Kosinus- und Sinusfunktionen ohne Phasenverschiebung – oder allein durch Sinusfunktionen mit Phasenverschiebung.
[2] Mit einer maximalen Auslenkung von 9% der Sprunghöhe.

wiederum ein Transformationspaar. Die *Umkehrfunktion der Diskreten Fouriertransformation* (Inverse Diskrete Fourier Transform, IDFT) berechnet sich fast analog und unterscheidet sich von der DFT in diesem Fall nur im Vorfaktor N^{-1} und dem Vorzeichen im Exponenten des Phasenteils in Gleichung 3.2.

Gleichung 3.2: Fourier-Transformationspaar für zeitdiskrete Signale [48, S.822]

$$\text{Analysegleichung (DFT):} \quad X(f_k) = \sum_{n=0}^{N-1} x(n) e^{-j2\pi n \frac{f_k}{f_A}} = X_{Re}(f) + jX_{Im}(f)$$

$$\text{mit } f_k = \frac{kf_A}{N} \Rightarrow \frac{f_k}{f_A} = \frac{k}{N}$$

unter Einhaltung des Abtasttheorems $f_A < 2f_{max}$

$$\text{Synthesegleichung (IDFT):} \quad x(n) = \frac{1}{N} \sum_{n=0}^{N-1} X(f_k) e^{j2\pi n \frac{f_k}{f_A}} \ ,$$

$$n, k = 0, \ldots, N-1$$

Gleichung 3.3 zeigt die Berechnung des Betrags- und Phasenspektrums aus N-Abtastwerten eines zeitdiskreten Signals $x[n]$. Die N-Abtastwerte entsprechen der Größe des meist glockenförmigen Analysefensters[1], das durch Multiplikation des Signals $x[n]$ mit einer Fensterfunktion $w[n]$ erzeugt wird. Ziel ist ein weiches Ein- und Ausblenden des Analysebereichs um die Anreicherung des Signals mit zusätzlichen Frequenzen, wie sie bei einer harten Rechteckfensterung entstehen, zu mindern.[48, S.819]

Gleichung 3.3: Diskretes Betrags- und Phasenspektrum

$$\text{Betragsspektrum:} \quad |X(f)|_{dB} = 20log(\sqrt{X_{Re}(f) + X_{Im}(f)}) \qquad [dB]$$

$$\text{Phasenspektrum:} \quad \varphi(f) = \arctan\left(\frac{X_{Im}(f)}{X_{Re}(f)}\right) \qquad [rad]$$

Liegt ein aperiodisches Signal vor oder ist die Periodendauer unbekannt, ist es möglich, mit Hilfe der Fensterung durch Aneinanderreihung des Fensterinhalts eine periodische Funktion $x'[n]$ zu erzeugen. Auf diese Weise erhält man bei der Transformation ein dem Spektrum des gefensterten Ausschnittes von $x[n]$ sehr ähnliches Spektrum $X'[f]$. Ist die zeitliche Veränderung eines Spektrums von Interesse, kann bei einer Fensterlänge von $N/2$ eine halbüberlappende gleitende Kurzzeit-Spektralanalyse über die gesamte interessierende Signallänge durchgeführt werden. [48, S. 820]

[1] Typische Fenster sind z. B. ein Hann- oder ein Hamming-Fenster.

Die Performanz eines Algorithmus' wird durch dessen Zeitkomplexität, d. h. der Anzahl der nötigen Rechenoperationen geschätzt. Dieses auch *asymptotische Laufzeit* genannte Entwicklungsverhalten kann im Vorfeld einer Implementierung unter Verwendung des nach Edmund Landau benannten *Landau Symbols*, auch bekannt als $\mathcal{O}$*-Notation* , unter der vereinfachenden Zusammenfassung ähnlich stark wachsender Teilkomponenten eines Algorithmus' abgeschätzt werden. In der Analysegleichung zu 3.2 ist ersichtlich, dass zur Berechnung einer Stelle im Frequenzbereich N Multiplikationen und $N-1$ Additionen nötig sind. Für N-Stellen mit $N \times N$ Multiplikationen und $N \times (N-1)$ Additionen geht die Zeitkomplexität der DFT gegen $\mathcal{O}(N^2)$.
Die FFT ist ein Sammelbegriff für geschickte Algorithmen zur effizienten Berechnung der DFT. Der bekannteste ist der einfach zu implementierende *Radix-2*-Algorithmus nach Cooley et al. [49] in Abb. 6, mit dem die DFT mit einem rechnerischen Aufwand von nur $\mathcal{O}(Nlog_2N)$ umsetzbar ist. [50, S.234]. Dies gelingt durch Ausnutzung von Symmetrien mit einer zusammenfassenden Umsortierung redundanter Rechenoperationen und der damit einhergehenden Reduzierung der Zahl der insgesamt benötigten Multiplikationen.[1] [47, S. 255 ff.] Diese Redundanzen kommen aufgrund von Symmetrien und Periodizitäten der Potenzen des komplexen Drehfaktors *(Twiddle Factor)* W_N zustande.[50] Die Periodizität gilt damit auch für die DFT/IDFT $X[k] = X[k+N] \Rightarrow x[n] = x[n+N]$. Eingesetzt in die nach dem *Danielson-Lanczos Lemma* (DL-Lemma) in gerade und ungerade Koeffizienten aufgespaltene Analysegleichung 3.2 führt dies zu:

Gleichung 3.4: DFT-Zerlegung mit Hilfe des DL-Lemma [111]

$$\begin{aligned} X(k) &= \sum_{n=0}^{N-1} x(n) W_N^{nk} \qquad \text{mit: } W_N = e^{-j\frac{2\pi}{N}} \\ &= \underbrace{\sum_{n=0}^{N/2-1} x(2n) e^{-j\frac{2\pi kn}{N/2}}}_{gerade(g)} + W_N^n \underbrace{\sum_{n=0}^{N/2-1} x(2n+1) e^{-j\frac{2\pi kn}{N/2}}}_{ungerade(u)} \\ &= X^g(k) + W_N^n X^u(k) \end{aligned}$$

Gleichung 3.4 zeigt den Beginn der DFT-Zerlegung mit einer Reduktion im Zeitbereich, welche eine von zwei Herangehensweisen bei der Reduzierung der Anzahl nötiger Multiplikationen der FFT ist. Der Unterschied zur alternativ möglichen und in Abb. 6 dargestellten DFT-Zerlegung mit Reduktion im Frequenzbereich besteht lediglich in der Art der Herleitung und der Struktur des Graphen, während die Eigenschaften und Ergebnisse äquivalent bleiben.[50, S. 235]
Bei einer Verschiebung des periodischen Zeitsignals um m-Punkte bleibt das Betragsspektrum gleich, lediglich der Phasenfaktor $e^{-j\omega m}$ muß mit der Fourier-Transformierten multipliziert werden. Dem obigen ersten Transformationspaar entspräche bei einer solchen zirkulären Verschiebung um m-Punkte das Paar $x_1[n-m]$ und $e^{-j\omega m}X[k_1]$. Die sich daher zum Teil wiederholenden Faktorberechnungen können deshalb, wie in Abb. 6 dargestellt, ressourcenschonend zusammengefasst werden. Vorrausgesetzt $N = 2^n$ mit $n \in \mathbb{N}$ kann eine sukzessive Aufspaltung der DFT, wie in

[1] Gilt ab einer 16-Punkte-FFT, darunter ist die Zahl benötigter Rechenoperationen bei DFT/FFT gleich.

Gleichung 3.4 skizziert, in $log_2 N$ Teil-DFT's mit jeweils $N/2$ Basisoperationen[1] solange erfolgen, bis eine vollständige Zerlegung in 2-Punkte-DFTs erfolgt ist. Für jede 2-Punkte-DFT sind 2 komplexe Multiplikationen und 2 komplexe Additionen notwendig. Eine Multiplikation und eine Addition zusammengefasst in einer Operation mit dem *Overlap-Add*- oder *Overlap-Save*-Verfahren ergibt N Operationen für jede Zerlegungsebene – daraus resultiert der Gesamtaufwand der FFT von $\mathcal{O}(Nlog_2 N)$.

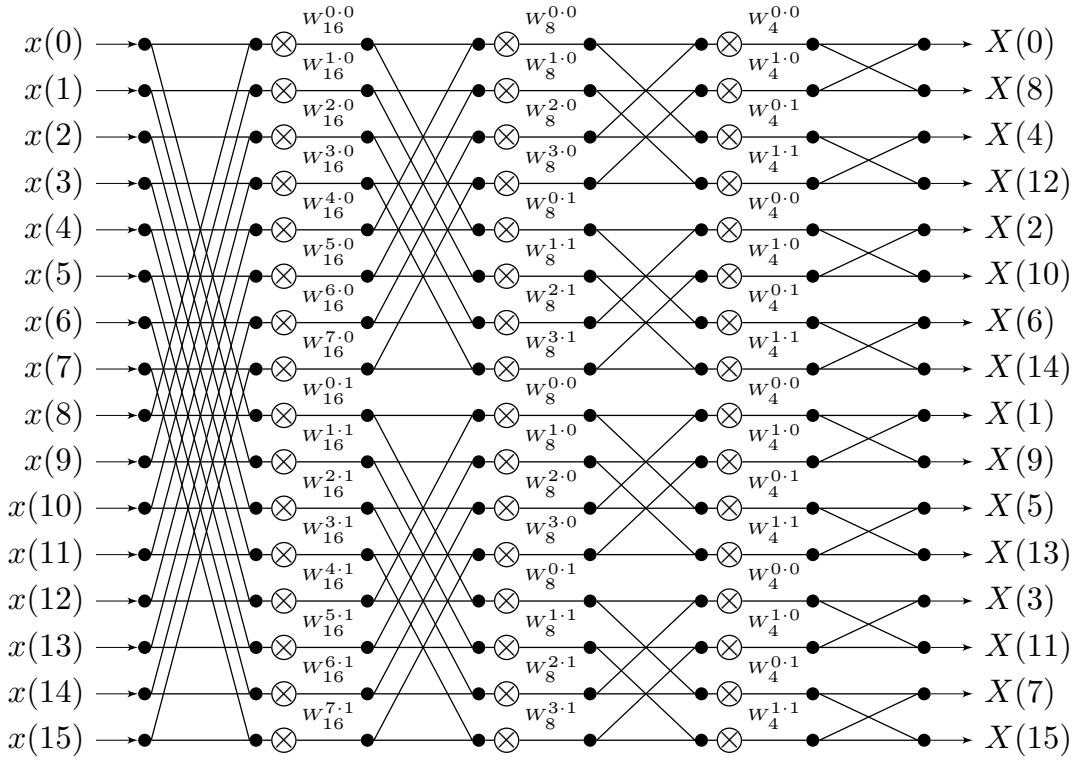


Abb. 6: Radix-2-FFT für 16 Elemente [112] mit Reduktion im Frequenzbereich [50, S. 244]

Für $N \neq 2^n$ kann ein sog. *Zero-Padding* durchgeführt werden, bei dem für die fehlenden Elemente bis zur nächsten Zweierpotenz Nullen eingefügt werden. Die Umsortierung der Elemente der Spektralfolge am Ende des Algorithmus' ist programmatisch durch eine einfache Invertierung der Binärzahl des Indexes k, einem *Bit-Reversal*[2], des Abtastwertes zu erreichen. Für eine detailliertere Beschreibung der FFT, insbesondere der DFT-Zerlegung sei z. B. an Smith [113] oder Kammeyer und Korschel [50] verwiesen – frühere Beschreibungen finden sich u. a. bei Cooley et al. [49], Weinstein [51] und nicht zuletzt bei Gauss [52]. Die Radix-2-FFT ist nur ein Beispiel einer ganzen Anzahl ähnlicher Algorithmen, wie z. B. dem Radix-4- bzw Radix-8-Algorithmus [50, S. 245], dem Primzahl-Algorithmus, der Chrip-z-Transformation, dem Goertzel- und dem Winograd-Algorithmus.

[1] Wird aufgrund der graphischen Struktur auch als „Butterfly“ bezeichnet.

[2] Bei $001 \xrightarrow{bitreversal} 100$ tauscht die 1 mit der 4.

3.1.2 Diskrete und Schnelle Faltung

Die Faltung ist die mathematische Operation, mit der in der BS die in den Übertragungsfunktionen enthaltenen Richtungs- und Raumeigenschaften auf ein nachhallfreies Audiosignal übertragen werden können. Dies ist nach Vorländer [53] bei Betrachtung des Schallausbreitungsraumes als LTI-System für den Fall der rückwirkungsfreien Signalübertragung möglich.
Die Faltung schiebt eine an der Ordinate gespiegelte Funktion $h[n]$ über eine andere Funktion $x[n]$ und berechnet währenddessen das durch die Elemente von $h[n]$ gewichtete Mittel. Dabei entsteht eine neue Funktion y[n]:

Gleichung 3.5: Diskrete Faltungssumme mit N Speicher [54, S.817]

$$y[n] = (x * h)[n] = x[n] * h[n] = \sum_{k=0}^{N-1} x[k] \cdot h[n-k]$$

Weitere Eigenschaften der Faltung sind Kommutativität, Distributivität, Assoziativität, Verzögerungsakkumulation und ihr neutrales Element ist δ_K. Die Faltung im Zeitbereich wird wegen ihres rechnerischen Aufwandes von $\mathcal{O}(N^2)$ für große N sehr „teuer“ [1]. Mit einer Multiplikation im Frequenzbereich gibt es unter Ausnutzung der Effizienz der FFT einen „kostengünstigeren“ Weg, diese Berechnung für große N durchzuführen.

Gleichung 3.6: Diskrete Faltung im Frequenzbereich

$$y[n] = x[n] * h[n] = IFFT(X(f)H(f))$$

Die Zeitkomplexität der sog. *Schnellen Faltung* im Frequenzbereich beträgt nach Orfanidis [55, S.521] unter der Annahme, dass das Spektrum $H(f)$ der *Impulsantwort* (Impulse Response, IR) $h[n]$ bereits vor der Laufzeit bekannt ist, $\mathcal{O}(2Nlog_2N + N)$ und ist somit für $N > 64$ im Vergleich zu 3.5 deutlich effizienter. Brigham [56] konstatiert daher ab der nächsthöheren Zweierpotenz $N = 128$ einen Effizienzvorteil auf Seiten der Schnellen Faltung. Gardner [57] setzt diese Werte für typische *Digitale Signalprozessoren (Digital Signal Processors)* (DSPs) bereits 1995 um eine Zweierpotenz niedriger an. Demnach gibt es den Effizienzvorteil bereits ab $N > 32$, was im Beispiel der Radix-2-FFT die nächsthöhere Zweierpotenz $N = 64$ wäre. Als vorteilhafter Nebeneffekt reduziert sich mit jeder eingesparten Operation auch die Zahl etwaiger Rundungsfehler, die ins Endergebnis mit eingehen. Bei der Berechnung der diskreten Faltung über eine Multiplikation im Frequenzbereich werden die Signale $x[n]$ mit der Länge L und $h[n]$ mit der Länge m mit Hilfe der FFT in ihre Frequenzspektren zerlegt, ihre Elemente miteinander multipliziert und anschließend mit der *Umkehrfunktion der Schnellen Fouriertransformation* (Inverse Fast Fourier Transform, IFFT) in den Zeitbereich zurück transformiert. Um bei der Verwendung der zirkularen Faltung im Frequenzbereich ein mit der aperiodischen Faltung übereinstimmendes Ergebnis

[1] Für jedes Sample sind N Additionen, N Multiplikationen und $2N$ Speicherzugriffe nötig. Bei einer Signallänge von M Samples, sind das 2MN Operationen und 2MN+1 Speicherzugriffe. Ist $M = N \Rightarrow \mathcal{O}(N^2)$.

für ein Ausgangssignal $y[n]$ mit der Länge $L + m - 1$ zu erhalten und Informationsverlust durch Überlappungen im Zeitbereich zu vermeiden, werden beide Signale durch Zero-Padding auf die Länge $N = L + m - 1$ gebracht. Anschließend erfolgt eine N-Punkte FFT beider Folgen [50, S. 249 f.][1], wobei sich die Größe von N nach dem verwendeten FFT-Algorithmus richtet – bei einer Radix-2-FFT gilt $N = L + m - 1 \leq 2^n$ mit $n \in \mathbb{N}$.[2] Sind die Frequenzspektren berechnet, ist der nächste Schritt eine punktweise Multiplikation ihrer Elemente. Abschließend gelangt man durch eine N-Punkte IFFT zurück zum Zeitsignal. Bei dieser Vorgehensweise muss die FFT für die gesamte Signaldauer berechnet werden bevor die Faltung durchgeführt werden kann, was zu einer Latenz von mindestens der Berechnungszeit von N Samples führt. Auch wäre bei dem in der Praxis häufig anzutreffenden Anwendungsfall einer Faltung eines langen Eingangssignals mit einer verhältnismäßig kurzen IR $L \gg m$ ein rechnerisch unvorteilhaft langes Zero-Padding notwendig.[50, S. 250] Für reale Anwendungen nutzt man daher die *Segmentierte Faltung*, bei der das Eingangssignal auf Grundlage der Linearität und Distributivität der Faltungsoperation in Blöcke unterteilt und stückweise mit der Impulsantwort gefaltet wird. Dafür gibt es zwei Berechnungsmethoden: das *Overlap-Add-Verfahren* (OLA) mit ausgangseitiger Überlappung der Segmentfaltungen und das *Overlap-Save-Verfahren* (OLS), bei dem eingangseitig sich überlappende Segmente in die Faltung geführt werden. Beim OLA wird die Eingangsfolge in eine Zahl $R = N/L$ nicht-überlappender Teilfolgen zerlegt und durch Zero-Padding auf eine Blocklänge von $N = L + m - 1$ gebracht.

Gleichung 3.7: OLA – Eingangsseitige Segmentierung in R Blöcke à L Samples [ebda.]

$$x[k] = \sum_{r=0}^{R-1} x_r[k], \qquad x_r[k] = 0 \begin{cases} \text{für} & k < r \cdot L \\ \text{und} & k \geq (r+1) \cdot L \end{cases}$$

Die Segmente werden separat N-Punkte fouriertransformiert, elementweise mit der Spektralfolge der IR[3] multipliziert und mittels N-Punkte IFFT zurück in ein Zeitsignal umgewandelt. Dabei entstehen Teilergebnisse der Länge $N = L + m - 1$ mit Überlappungsbereichen zwischen $y_r[k]$ und $y_{r-1}[k]$ der Länge $m - 1$, welche auch das Ausklingverhalten des jeweils vorangegangenen Blocks enthalten. Durch Superposition der Teilfaltungen mit punktweiser Addition der übereinanderliegenden Werte generiert das OLA eine fortlaufende Ausgangsfolge, die der aperiodischen Faltung entspricht.

Gleichung 3.8: OLA – Ausgangsseitige Superposition [ebda.]

$$y[k] = \sum_{r=0}^{R-1} y_r[k], \qquad y_r[k] = 0 \begin{cases} \text{für} & k < r \cdot L \\ \text{und} & k \geq (r+1) \cdot L + m - 1 \end{cases}$$

[1] Kammeyer und Korschel [50] definieren die Länge der IR mit $m + 1 \Rightarrow N = L + m$.

[2] Für $N \neq 2^n$ mit $n \in \mathbb{N}$ wird das Zero-Padding bis zur nächsthöheren Zweierpotenz fortgesetzt.

[3] Eine Segmentierung der IR bei $m > L$ erfolgt analog zur Segmentierung von $x[n]$. Findet keine Segmentierung statt, ist Zero-Padding auf eine Länge von $N = L + m - 1$ notwendig.

Das OLS zerlegt hingegen bereits die Eingangsfolge in überlappende Segmente und berechnet daraus die zirkulare Faltung der Teilfolgen mit der IR. Zu Beginn muss aufgrund der Periodizität ein Zero-Padding mit $m - 1$ Nullen erfolgen um Aliasingfehler zu vermeiden. Aufgrund der eingangsseitigen Überlappung erhält man am Ausgang zwar mehr Werte, als für die Konstruktion des gewünschten Ausgangssignals notwendig sind, jedoch stimmen diese durch die Einbeziehung des jeweiligen Vorgängersegments in Teilbereichen bereits direkt mit dem Werteverlauf der linearen Faltung überein. Diese $m - 1 \leq n \leq N - 1$ Teilbereiche können daher einfach ausgeschnitten und zusammengesetzt werden, während die ersten $m - 1$ Werte als Aliasingfehler verworfen werden. Das fortlaufende Ausgangssignal ergibt sich durch Addition der so gefilterten Teilergebnisse. [50, S.258 f.] Durch den Wegfall von $(R - 1)(m - 1)$ Additionen der Überlagerungsbereiche der Teilfaltungen ist das OLS geringfügig "günstiger" als das OLA. Beide Verfahren sind ähnlich aufwändig und beinhalten eine systembedingte Latenz in Höhe der Berechnungszeit für die ersten N Samples, die die Länge der IR m als feste Größe enthält. Häufig hat man es in der Virtuellen Akustik mit IRs mit relativ großem m im Verhältnis zur gewünschten Blockgröße zu tun. Daher ist es naheliegend auch diese mittels OLA oder OLS blockweise in die Faltung zu führen. Eine weitere Optimierung ganz ohne Latenz bietet Gardner [57] mit seinem *Zero-Delay Convolution Algorithm* an. Sein Algorithmus kombiniert die direkte Faltung in Form eines FIR-Filters für die Berechnung der ersten Ausgangswerte mit der Partitionierten Schnellen Faltung für die darauffolgenden. Als FIR-Filter bezeichnet man ein *System mit endlicher Impulsantwort* (Finite Impulse Response, FIR). Die Blockgröße für die ersten Segmente der Schnellen Faltung wählt Gardner mit $N = 32$ oder $N = 64$ [ebda.] so aus, dass bereits ein leichter Effizienzvorteil gegenüber der direkten Faltung besteht. Zusätzlich soll die Segmentlänge N mit zunehmenden k wachsen um die Effizienz der FFT voll auszunutzen. Damit sollen sofort Ausgangswerte zur Verfügung stehen, die Prozessorlast jedoch mit der Zeit sinken. Allerdings ist genau diese anfänglich sehr hoch, da auch nach dem Wechsel von der direkten Faltung auf die Segmentierte Faltung deren wachsende Blöcke zu Beginn nahezu zeitgleich berechnet werden müssen um eine fortlaufende Ausgabe der Systemantwort zu gewährleisten. Zudem werden variable Blocklängen in Objective-C und in der in dieser Arbeit verwendeten Echtzeitentwicklungsumgebung *Pure Data* nicht nativ unterstützt. Um die Kommunikation zwischen App und Betriebssystem einfach zu gestalten wird daher auf eine Faltungsmethode mit konstanter Blockgrösse zurückgegriffen.

3.2 Dynamische Binauralsynthese

Der äußere menschliche Hörapparat kann hinsichtlich seiner Richtungsabhängigkeit bei der Schallübertragung als *Lineares Zeitinvariantes System* (Linear Time Invariant System, LTI-System) gesehen und damit über seine IR eindeutig beschrieben werden. Die linearen Verzerrungen, durch die ein Schallsignal bei der Übertragung im Freifeld durch die Geometrie von Torso, Kopf und Außenohr moduliert wird, sind in den HRTFs für die jeweilige Richtung als Modell des physikalischen Übertragungsweges kodiert und können durch Faltung mit einem nachhallfreien Audiosignal[1] auf dieses abgebildet werden. Da der Luftschall mit einer um 40 dB niedrigeren Hörschwelle

[1] Nicht-nachhallfreie Audiosignale tragen bereits Rauminformationen in sich. Diese würden gleichfalls mit in die Berechnungen einfließen und es käme zu einem verwirrenden Mix der Informationen des enthaltenen und des

als der Knochenschall die akustische Wahrnehmung dominiert [14, S. 89], wird letzterer bei diesem Modell vernachlässigt. Durch Austausch der HRTFs kann eine Änderung der wahrgenommenen virtuellen Schallquellenposition beim Hörer bewirkt werden. Den dynamischen Austausch der HRTFs unter Einbeziehung der aktuellen Kopfbewegungen des Hörers in Echtzeit nennt man eine *Dynamische Binauralsynthese* (Dynamic Binaural Synthesis, DBS). Ziel ist dabei der Erhalt der wahrgenommenen Klangquellenposition. Grundlage dieses Konzepts ist der kopfbezogene Ansatz der binauralen Tonaufnahme in der Kunstkopfstereophonie.

Binaural recording: *„The idea behind the binaural recording technique is as follows: The input to the hearing consists of two signals: sound pressures at each of the eardrums. If these are recorded in the ears of a listener and reproduced exactly as they were, then the complete auditive experience is assumed to be reproduced, including timbre and spatial aspects. The term binaural recording refers to the fact that the two inputs to the hearing are reproduced correctly.“*
Møller [58, S. 171f]

Während dabei die realen physikalischen Parameter der Schallausbreitung von der Quelle bis zum Trommelfell direkt in die Tonaufnahme eingehen, versucht die BS diese algorithmisch zu simulieren. Dazu wird ein Paar BF geschaffen, das die richtungsabhängigen Eigenschaften des Signalweges zu dem der Schallquelle zugewandten *ipsilateralem* Ohr und zu dem von ihr abgewandten *kontralateralem* Ohr enthält. Auf diese Weise wird auf Grundlage der Linearität das Modell der Schallquelle mit dem Modell des Übertragungsweges zu einer möglichst genauen Approximation des natürlichen, räumlichen Signals am Empfangsort zusammengefügt. Sind nur die Richtungsinformationen gefragt, kann die Auralisation mittels HRTFs erfolgen, welche im Freifeld generiert werden – eine andere Möglichkeit ist die Verwendung *Binaurale Raumimpulsantworten (Binaural Room Responses)* (BRIRs) in denen zusätzlich zu den Richtungs- auch Rauminformationen enthalten sind.

3.2.1 Akustische Richtungssimulation mit HRTFs

Zur Generierung der HRTFs kann ein nach Durchschnittsmaßen geformter *Kunstkopf* (Dummy-Head, DH) oder eine Versuchsperson verwendet werden. Dementsprechend werden individuelle und nicht-individuelle HRTFs unterschieden. Eine weitere Möglichkeit der Erstellung nicht-individueller HRTFs ist die Mittelung eines Datensatzes individueller HRTFs, welche jedoch zu unerwünschten Verfälschungen führen kann. Wie in Abbildung 1 dargestellt, ist die HRTF abhängig von den Winkeln Azimut φ, Elevation θ und dem Abstand zwischen Schallquelle und Empfänger r, wobei der Einfluss von r außerhalb des unmittelbaren Nahfeldes bereits in einem Abstand $r > 2$ m stark nachlässt [58, S. 179] und ab dort eher als Parameter der Sound-Externalisation zu verstehen ist [59, S. 30].

abgebildeten Raums.

Nach Blauert sind an der individuellen Formung der Ohrsignale neben dem Außenohr auch – abnehmend in ihrer Wichtigkeit bzgl. ihres richtungsabhängigen Einflusses auf die Ohrsignale – Kopf-, Schulter- und Rumpf mit Beugungs-, Streuungs- und Reflexionsprozessen beteiligt [60].
Je nach Domain unterscheidet man die *Kopfbezogene Impulsantwort* (Head Related Impulse Response, HRIR) im Zeitbereich und die HRTF im Frequenzbereich. Für die Bestimmung der HRTF wird der signalverändernde Einfluss der Körpergeometrie in Abhängigkeit der oben genannten Parameter ermittelt. Dabei kann der weitgehend richtungsunabhängige Einfluß des Hörkanals und der Trommelfellimpedanz ein- bzw. ausgeschlossen werden.[1]

Gleichung 3.9: Bestimmung der Außenohrübertragungsfunktion nach Blauert [61] zitiert in Møller [58, S. 179]

$$\frac{P(f)_4}{P(f)_1} = \frac{\text{Schalldruckspektrum am Trommelfell}}{\text{Schalldruckspektrum am Referenzpunkt in Abwesenheit des DHs/Hörers}}$$

$$\frac{P(f)_3}{P(f)_1} = \frac{\text{Schalldruckspektrum am offenen Ohreingang}}{\text{Schalldruckspektrum am Referenzpunkt in Abwesenheit des DHs/Hörers}}$$

$$\frac{P(f)_2}{P(f)_1} = \frac{\text{Schalldruckspektrum am geblockten Ohreingang}}{\text{Schalldruckspektrum am Referenzpunkt in Abwesenheit des DHs/Hörers}}$$

Nach der normalisierenden Division durch das Amplitudenspektrum des Referenzpunktes zur Minimierung des Amplitudenbereichs definiert sich die monoaurale Freifeldaußenohrübertragungsfunktion:

$$HRTF_{\text{mono_FF}} = \frac{(P(f)_i/P(f)_1)(\varphi, \theta, r)}{(P(f)_i/P(f)_1)(\varphi = \theta = 0°)},$$

die monoaurale Diffusfeld-Außenohrübertragungsfunktion:

$$HRTF_{\text{mono_DF}} = \frac{(P(f)_i/P(f)_1)(\varphi, \theta, r)}{|P(f)_i/P(f)_1|_{\text{alle Winkel}}},$$

[1] Vgl. Gleichung 3.9

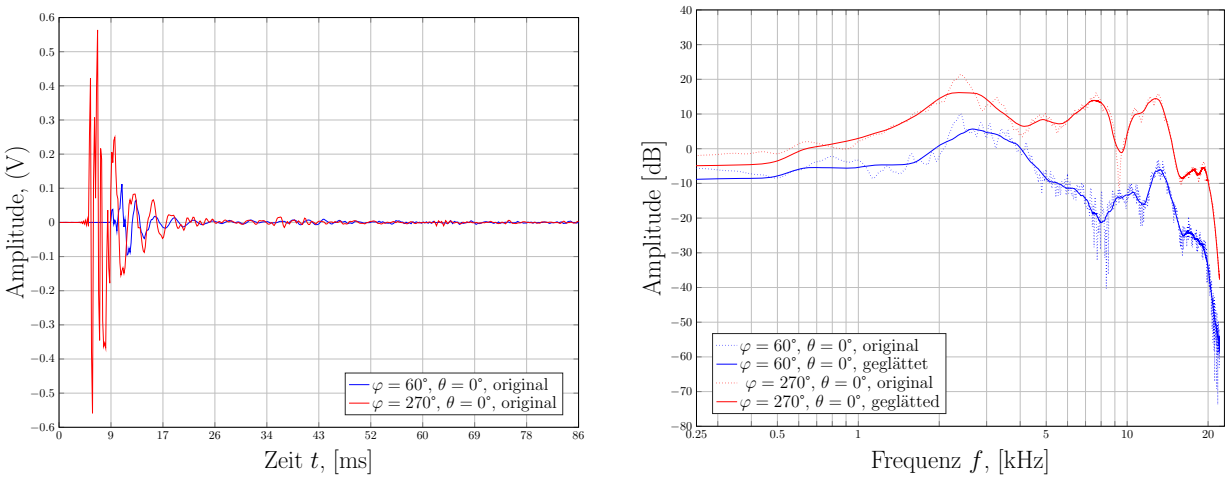
und die Unterschiede zwischen den Ohrsignalen werden durch die interaurale Außenohrübertragungsfunktion abgebildet:

$$\begin{aligned} HRTF_{\text{inter_FF}} &= \frac{(P(f)_i/P(f)_1)_{\text{kontralaterales Ohr}}}{(P(f)_i/P(f)_1)_{\text{ipsylaterales Ohr}}}(\varphi,\theta,r) \\ &= \frac{(P(f)_i)_{\text{kontralaterales Ohr}}}{(P(f)_i)_{\text{ipsylaterales Ohr}}}(\varphi,\theta,r). \end{aligned}$$

$$i = \text{ beliebiger Meßpunkt } 2-4$$

Für die Messung mit dem Kunstkopf wird meist der Messpunkt am geblockten Ohr bevorzugt, da bereits hier alle relevanten Richtungsinformationen vorliegen. Das Mikrofonsignal am DH (oder den Ohrmikrofonen der Versuchsperson) unterliegt bei der HRTF-Erstellung auch den spektralen Verzerrungen durch die Lautsprecher- und Mikrofonübertragungsfunktion, welche abschließend aus den HRTF-Rohdaten entfernt werden müssen. Ein kompakter Überblick der nötigen Teilschritte zur Generierung der korrigierten HRTF findet sich bei Fellner und Höldrich [60], ausführliche Abhandlungen zu diesem Thema bei Møller [58] oder Begault [62]. Hochfrequente spektrale Dynamik unterhalb der durch die kritischen Bänder bestimmten JND, kann durch Funktionsglättung entfernt werden Nicol [59, S. 31].

Abb. 7: Kopfbezogene Impulsantwort und Außenohrübertragungsfunktion, MIT Kemar [114]



(a) HRIR, linkes Ohr

(b) HRTF, linkes Ohr

Die HRTFs in Abbildung 7b sind jeweils um 60° aus der Medianebene ausgelenkt. Neben der Funktionsglättung sind deutlich auch die richtungsspezifischen Parameter der BS in Form spektraler Verzerrungen, ILDs und ITDs zwischen ipsilateralem (rot) und kontralateralem Ohr (blau) zu erkennen.
In dieser Arbeit wird vereinfachend von einer idealen Kopfsymmetrie ausgegangen. Deshalb kann nach Formel 3.10 [58, S. 181] die Hälfte des Speicherplatzes eingespart werden.

Gleichung 3.10: Reduzierung des HRTF-Datensatzes

$$\frac{P(f)_{i,\text{links}}}{P(f)_1}(\varphi,\theta,r) = \frac{P(f)_{i,\text{rechts}}}{P(f)_1}(-\varphi,\theta,r)$$

Um den Datensatz zusätzlich zu verkleinern, ist es möglich ein HRTF-Raster mit hinreichend feiner Auflösung anzulegen und die Positionen zwischen den vorliegenden HRTF-Stützpunkten zu interpolieren. Eine weitere auch in dieser Arbeit aufgegriffene Möglichkeit der Optimierung im Hinblick auf ressourcenschonende Signalverarbeitung in Echtzeitanwendungen ist eine drastische Verkürzung der Filterlängen. In Abbildung 7a ist ersichtlich, dass der Hauptanteil der Dynamik im vorderen zeitlichen Drittel der Audiodatei stattfindet. Tatsächlich ist es möglich die BFs unter weitgehender Beibehaltung der Lokalisierungsleistung nach Xiang et al. [23, S. 2] auf 128 und nach Begault [62, S. 128 ff.] mit einer Rechteckfensterung dieses Bereiches mit dem höchsten Informationsgehalt sogar auf 64 Samples (Taps) zu verkürzen. Einerseits stellt die Verwendung nicht-individueller HRTFs in 3D-Audio-Applikationen hinsichtlich der Qualität der Auralisation im Vergleich zur Verwendung indivdueller HRTFs einen Kompromiss zugunsten der Umsetzbarkeit dar. So kann bei geringeren Kopfmaßen im Vergleich zum DH bei horizontaler Bewegung die virtuelle Schallquelle in die Gegenrichtung wandern [63] oder es können Elevations-, Entfernungsfehler [ebda] oder eine FBA auftreten [15]. Andererseits besteht nach Seeber [15] die Möglichkeit auditiver Lernprozesse durch die “Adaption an veränderte interaurale Merkmale“ [ebda., S. 106].

3.2.2 Akustische Raumsimulation mit (B)RIRs

Methoden der geometrischen Akustik zur Raumsimulation, wie das Spiegelschallquellenmodell [64] oder das Raytraycingverfahren [65], können je nach Grad der Auflösung (und/oder Zahl der Quellen) enorm rechenintensiv werden. Der hohen Plausibilität, die mit diesen Methoden erreicht werden kann, stehen neben dem Echtzeitanspruch auch die begrenzten Ressourcen eines akkubetriebenen Smartphones gegenüber, weshalb hier zu einem vereinfachenden Algorithmus gegriffen werden muss.
Bei alleiniger Verwendung von HRTFs ist aufgrund des Freifeldcharakters keine Abbildung von Rauminformationen möglich. Werden BFs in natürlichen Räumen akquiriert erhält man BRIRs, die sämtliche Reflexionen des Raumes enthalten, dadurch jedoch je nach Nachhallzeit sehr viel höhere Filterlängen aufweisen. Eine weitere Einschränkung ist die Rarität von BRIRs für Außenumgebungen und der hohe Zeitaufwand, den die Erstellung einer ganzen Anzahl prototypischer BRIR-Datensätze mit sich bringt. Auch der interessante Ansatz der Vor-Ort-Akquise mittels Klatschimpuls von Gamper und Lokki [66] bietet für die geplante Anwendung keine Lösung. Ein Kompromiss ist durch eine vorangehende Faltung des Audiosignals mit einer monoauralen OIR und anschließender HRTF-Faltung möglich.

3.2.3 HpTF-Kompensation

Grundlegend für das Gelingen einer binauralen Simulation ist die exakte Reproduktion der Schalldrücke am Trommelfell [58, S. 171]. Bei der Kopfhörerwiedergabe sind jedoch Teilstrecken des

Signalweges doppelt im Wiedergabesignal enthalten [67, S. 283], einmal durch die HRTF des Kunstkopfes und durch Teile der HRTF des Hörers. Für das Beispiel circum auraler *Kopfhörer* (Headphone, HP) beginnt die Dopplung bei den Pinna-Resonanzen und setzt sich je nach Messpunkt der HRTF über Ohrkanal- bis zu Trommelfellresonanzen fort. Hörversuche zeigten, dass spektrale Verzerrungen dem oben genannten Ziel entgegenwirken [68] und daher müssen die durch die Aufnahme- und Wiedergabetechnik hervorgerufenen zusätzlichen Signalmodifikationen kompensiert werden. Dies gilt insbesondere für die Verwendung nicht-individueller HRTFs, die mit einer höheren Fehlerquote bei der Synthese binauraler Ohrsignale und damit in der räumlichen Abbildung einhergehen kann [15, S. 60]. Für die aufnahmeseitig verwendeten Lautsprecher und Mikrofone bei der HRTF-Generierung ist deren Frequenzgangkompensation bereits Teil des Prozesses [60] – die *Kopfhörerübertragungsfunktion* (Headphone Transfer Function, HpTF) ist jedoch zu diesem Zeitpunkt oft noch nicht bekannt, hängt u. a. auch von der Methode der HRTF-Aufnahme ab und muss daher am gleichen Referenzpunkt generiert werden [58]. Ist die HpTF bekannt, dient diese der Suche eines geeigneten FIR-Filters mit einer Zielfunktion, die zusammen mit der HpTF in einem neutralem Element mündet [69, S. 33].

Gleichung 3.11: Bedingungen an HpTF-Entzerrfilter

Frequenzbereich: $$H_{eq}(\omega) = H(\omega) \cdot H_c(\omega) = 1$$

Zeitbereich: $$h_{eq}(n) = \sum_{k=0}^{N-1} h(k)h_c(n-k) = h(n) * h_c(n) = \delta(n)$$

Die Kompensation der HpTF kann über eine Filterung des Audiosignals auf Grundlage der Inversen der komplexen Übertragungsfunktion erfolgen. [68, S. 519].

Gleichung 3.12: HpTF-Kompensationsfilter

$$H_C(z) = H(z)^{-1}$$

Dabei ist die HpTF-Kompensation nicht als trivial einzustufen, wie es nach der Gleichung 3.12 den Anschein haben mag. Schärer und Lindau [68, S. 519 ff.] beschreiben weiter, wie durch verschiedenen Kompensationsverfahren mit den Problemen, wie den gemischtphasigen Eigenschaften der meisten Systeme, dem Bandpasscharakter der meisten realen akustischen Wandler oder der Gebundenheit der Übertragungsfunktion an eine feste Konstellation umgegangen werden kann. Detaillierte Beschreibungen der Vorgehensweise bei der HpTF-Entzerrung finden sich u. a. auch bei Møller [58] und Lindau und Brinkmann [70]. Bei der Kopfhörerwiedergabe entfällt neben der *Lautsprecherübertragungsfunktion* (Loudspeaker Transfer Function, LsTF) auch der Einfluß des Raumes oder der Körpergeometrie auf die Ohrsignale. Da Kopfhörer jedoch meist für die Wiedergabe von im Diffusfeld entstandenen Audioaufnehmen verwendet werden, wird deren Übertragungsfunktion bereits konstruktionstechnisch für dieses angepasst. Theile [71] schlägt auf Grundlage seines

Assoziationsmodells zur Angleichung der Kopfhörerwiedergabe an die raumbezogene Lautsprecherwiedergabe und zum Zwecke der Externalisierung des Hörerlebnisses eine Diffusfeldentzerrung vor, wobei als Zielfunktion eine aus allen Einfallsrichtungen gemittelte HRTF vorgesehen ist. Das Bezugsschallfeld einer auf die Art gestalteten Zielfunktion ist das Diffusschallfeld und das zugehörige Verfahren ist in ITU-R BS.708 [4] formuliert und für viele hochwertige Studiokopfhörer internationaler Standard [72, S. 350]. Dies stellt eine zusätzliche, unerwünschte Dopplung und Verfälschung binauraler Informationen dar [67, S. 283]. Der Vollständigkeit zuliebe soll auch erwähnt werden, dass es durchaus auch Kopfhörer mit Freifeldentzerrung gibt, welche jedoch relativ selten anzutreffen sind. Die oben genannte Zielfunktion, an die sich die Hersteller mehr oder weniger halten (können), steht somit im Widerspruch zu der von Møller [58] für die Wiedergabe binauraler Aufnahmen empfohlenen möglichst linearen, flachen HpTF. Im vorliegenden Anwendungsfall ist der Kopfhörertyp des App-Nutzers oder der App-Nutzerin unbekannt und vielzählig, so dass ein Set von HpTFs mit einem erheblichen Umfang benötigt würde. Zudem variiert die HpTF selbst eines Kopfhörermodells mit der Sitzposition, was mit der starren Entzerrfunktion durch unerwünschte konstruktive Überlagerung zu Klingelartefakten führen kann [68, S. 519 ff]. Diese betreffen zwar den gesamten Frequenzbereich, sind jedoch im oberen Frequenzbereich stärker [ebda.]. Dem kann durch eine Mittelung von mehreren HpTF-Messergebnissen oder durch eine stärkere Berücksichtigung der Peaks, kombiniert mit einer geringeren Kompensation der Notches begegnet werden [ebda.]. Diese Vorgehensweise setzt allerdings die Kenntnis HpTF voraus, daher soll hier anstelle einer exemplarischen individuellen HpTFs-Entzerrung für ein einzelnes Kopfhörermodell im Sinne einer breiteren Anwendbarkeit ein reduzierter Entzerrfilter verwendet werden. Dieser belässt den besonders stark variierenden hochfrequenten Bereich ab 6 kHz [73, S. 223] der HpTF unverändert und nimmt eine Kompensation des Frequenzganges nur im Bereich des charakteristischen Peaks im Bereich um 3 kHz [73, S. 229], [74] vor. Diese Anhebung ist das Ergebnis der Summe von Außenohrresonanzen und auch in Abbildung 7 zu sehen.

3.2.4 Headtracking

Der dynamische Austausch der HRTFs synchronisiert mit den Kopfbewegungen des Hörers ist definierter Bestandteil der DBS und erfolgt über ein Head-Tracking-System, das in Echtzeit Updates der aktuellen Kopfposition an den Auralisationsalgorithmus schickt um dort sofort Eingang in den Audiosyntheseprozess zu finden. Für die Bewegungserfassung kann Ultraschall, Trägheitseffekte, Optik, magnetische [59, S. 42] oder elektromagnetische [75] Felder genutzt werden, wobei die Latenz der Sensoren, deren Messgenauigkeit (Drift) und Störanfälligkeit die Qualität der DBS entscheidend mitbestimmen. Da in dieser Arbeit kein externer Headtracker eingesetzt und stattdessen auf die vorhandenen Trägheits- und Magnetfeldsensoren des Smartphones zurückgegriffen wird, soll an dieser Stelle zu diesem Thema auf weiterführende Literatur resp. Dörner et al. [7] und Vorländer [9] verwiesen werden. Die Methoden und Probleme der Kopfbewegungserfassung im Rahmen der hier zu erstellenden App werden im Kapitel 4.5 besprochen.

3.3 Augmented Reality

Sutherland, 1965: *„A display connected to a digital computer gives us a chance to gain familiarity with concepts not realizable in the physical world. It is a looking glass into a mathematical wonderland. [. . .] The ultimate display would, of course, be a room within which the computer can control the existence of matter. A chair displayed in such a room would be good enough to sit in. Handcuffs displayed in such a room would be confining, and a bullet displayed in such a room would be fatal.“* [6]

Der Begriff AR taucht das erste mal 1955 bei Morton Heilig in seiner visionären Beschreibung eines "Kinos der Zukunft" auf – er nannte es *Sensorama*. Mittels digitaler Signalverarbeitung sollten bereits damals in diesem Kino alle Sinne angesprochen werden.[76, S. 4f.] Obwohl man bei der Vorstellung einer computergestützten Erweiterung der menschlichen Erfahrungswelt anfangs visuelle Virtualisierungen fokussierte, zeigt auch das obige Zitat farbenreich, dass es immer auch um die Einbeziehung weiterer Sinne ging. Bei Furht [76] und Dörner et al. [7] finden sich sehr ausführliche Abhandlungen zu dem Begriff. Die AR ist eng an die *Virtuelle Realität* (Virtual Reality, VR) geknüpft. Letztere macht dem Nutzer oder der Nutzerin über seinen oder ihren visuellen, auditiven, taktilen oder olfaktorischen Sinn eine künstliche Welt in einer Weise interaktiv erfahrbar, bei der die reale Welt so weit wie möglich ausgeklammert bleibt Die AR nutzt ähnliche Techniken, jedoch befinden sich reale und virtuelle Umgebung nicht in verschiedenen Kontexten, stattdessen dienen hier die virtuellen, computergenerierten Informationen einer Erweiterung der realen Erlebniswelt. Die damit steigenden Anforderungen hinsichtlich Tracking und Registrierung benannte Azuma [8] als Charakteristika für ein *System für Augmentierte Realität* (Augmented Reality System, ARS):

(a) Kombination realer und virtueller Elemente

(b) interaktive Steuerung in Echtzeit

(c) stimmige Registrierung der Elemente beider Welten in 3D

Registrierung meint hier die korrekte Einpassung der virtuellen Elemente in die reale Umgebung.[7, S. 264] Eine systematische Einordnung zeigt Abbildung 8 mit dem (Virtual Reality continuum, VRC) nach Milgrim et al. [77]. Zwischen der realen und virtuellen Umgebung als dessen beiden Randwerten befindet sich die *Gemischte Realität* (Mixed Reality, MR). Diese beinhaltet ineinander übergehend die *Augmentierte Virtualität* (Augmented Virtuality, AV) mit Fokus auf die virtuelle Umgebung und die AR in Richtung der realen Umgebung.

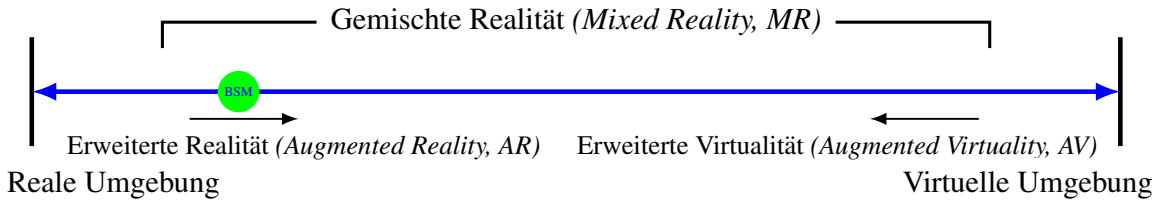


Abb. 8: Verortung der App *„Binaural Soundmap "* im Realität-Virtualitäts-Kontinuum nach Milgrim et al. [77]

Der Begriff *Augmentierte Virtualität* ist in der Praxis nur selten anzutreffen und bezeichnet einen eher theoretischen Sachverhalt. Die Einbeziehung und Registrierung der realen Welt als Unterschied zwischen AR und VR wird am Beispiel der technischen Umsetzung videobasierter Simulation für stereoskopische Bilder schnell deutlich: Während die *Videobrillen (Head Mounted Displays)* (HMDs) für VR wie die *Oculus Rift*™ einen undurchsichtigen Bildschirm aufweisen, ist dieser bei AR-unterstützenden HMDs wie der *VR-One*™ (wahlweise) lichtdurchlässig, damit die Außenwelt weiterhin wahrnehmbar bleibt und virtuelle Objekte in dieser registriert werden können. Die *Akustisch Augmentierte Realität* (Augmented Reality Audio, ARA) existiert im Zusammenspiel mit visuellen Augmentierungen, in Form einer *Mensch-Computer-Schnittstelle* (Human Computer Interface, HCI), sowie in Alleinstellung als Anwendung mit rein akustischen Augmentierungen. Während diese Augmentierungen per Definition computergeneriert sind, gibt es Punkt (a) aus obigen AR-Anforderungen betreffend für die Übertragung der akustischen Außenumgebung zwei Möglichkeiten. Im mittleren Bereich der MR bewegt man sich mit der Hear-Through-Methode, bei der die reale Soundscape mit z. B. in den Kopfhörern installierten Mikrofonen aufgenommen und vom Computer zusammen mit AR-Inhalten gemischt werden. Tönges [78] nutzt diese Hear-Through-Methode bei seiner Anwendung für virtuell spatialisiertes Up-Mixing von Stereo-Musikdateien mit anschließender BS, bei der die auf diese Weise akustisch augmentierten Inhalte variabel in die reale akustische Umgebung integriert werden können.[ebda.]
Der Anwendungsbereich der zu implementierenden MARA-Anwendung sind urbane Außenumgebungen, bei denen eine Ausblendung realer akustischer Ereignisse verkehrstechnisch ein Sicherheitsrisiko darstellen könnte. Dieses und die Einordnung der App in den linken äußeren AR-Bereich der in Abbildung 8 dargestellten MR sprechen für die alternative Möglichkeit der passiven Übertragung des Reality-Kanals über akustisch besonders durchlässige, circumaurale offene Kopfhörer. Die augmentierten Audioinhalte können in die reale akustische Umgebung ein- und ausgeblendet werden, während die Wahrnehmung der realen akustischen Umgebung so wenig wie möglich verändert wird. Damit ist Punkt (a) der von Azuma [8] formulierten Anforderungen an eine AR-Anwendung erfüllt – Punkt (b) und (c), die interaktive Steuerung und die Forderung nach perspektivischer Registrierung virtueller Inhalte wird durch geolokalisiertes Head-Tracking und durch Anwendung der DBS entsprochen.

4 Methoden: Die App *Binaural-Sound-Map*

4.1 Konzept: Idee, Features und Umsetzung

Ziel der Arbeit ist die Implementierung einer mobilen Demonstrations-App für geolokalisierte DBS, mit der virtuelle Schallquellen arbiträr geografisch positioniert, modifiziert, verwaltet und auralisiert werden können. Somit kann die App in der Sparte für *Ortsgebundende Datenverarbeitung und Dienste* (Location-based Computing and Services, LCS) für MARA angesiedelt werden – im Folgenden kurz *Ortsgebundenes Augmented Audio* (Location-based Augmented Audio, LAA). Als Anwendungsbereiche sind z. B. entdeckungsorientierte Stadtführungen, interaktive (Hör-)Spiele, Lernanwendungen oder Infotainment vorstellbar.
Es soll eine Datenbankstruktur geschaffen werden, in der eine Soundwalk-Entität in einer One-To-Many-Beziehung die thematisch ihr zugehörigen Soundspot-Entitäten aufnimmt. Zu den Attributen der Soundwalk-Entität gehören Metadaten wie der thematische Kontext, die grobe geografische Lage und die zugeordneten Soundspots. Diese verfügen selbst über Attribute wie Audioinhalt, prototypische akustische Szene in Form der OIR, Geofence, geografische und postalische Ortsdaten und weitere kontextbezogene Informationen. Über die *Grafische Benutzerschnittstelle* (Graphical User Interface, GUI) sollen die Attribute dieser beiden Entitäten inhaltlich modifiziert werden und die Soundwalks samt ihrer Soundspots in einer auf dem Smartphone abgelegten Datenbank verwaltet werden können. Da die Software auf einem Gerät mit begrenzten und geteilten Ressourcen laufen soll, kann von dieser keine physikalisch korrekte Schallfeldrekonstruktion erwartet werden. Vielmehr ist eine Kombination aus psychoakustischen Simulationen der für das Richtungs- und Entfernungshören wichtigsten Cues und eine unterstützende sounddesignerische Hervorhebung einzelner Effekte ein realistischeres Ziel.
Eingrenzend soll die App ohne zusätzliche Hardware in Form eines Head-Trackers lauffähig sein, d. h. dass für die Akquise der Kopfpositionsdaten die im Smartphone bereits vorhandenen Sensoren verwendet werden. Das bedeutet für das GUI, dass der Blick auf das Display während der laufenden Anwendung nicht notwendig sein soll, da das Smartphone – mittig auf den Kopfhörerbügeln befestigt – die Funktion des Head-Trackers erfüllt. Perspektivisch soll die App zusammen mit einem „intelligenten", sensorbestückten Kopfhörer verwendet werden. Daher soll die GUI eine grafische Anzeige der Position des Nutzers oder der Nutzerin und der Lage der Soundspots enthalten. Als weitere Eingrenzung wird festgelegt, dass die mobile Anwendung zur Laufzeit jeweils eine ruhende Quelle auralisieren soll. Die Software soll dabei selbständig und per Nearest-Neighbor-Berechnung die relevante virtuelle Schallquelle bei der Überquerung eines Geofences aktivieren und mit der Auralisation beginnen.
Die Auralisation soll Bewegungen in der in Abbildung 1 als Parameter des Richtungshörens dargestellten Azimut- und Elevationswinkel sowie die Entfernung von der virtuellen Schallquelle berücksichtigen. Für die richtungsspezifische Simulation ist die Verwendung eines auf 128 Samples [23] gekürzten HRTF-Datensatzes [114] vorgesehen. Die zu simulierende akustische Umgebung befindet sich im teilbegrenzten, urbanen Halbfreifeld, von dem vereinfachend angenommen wird, dass sich in unmittelbarer Nähe zur virtuellen Schallquelle keine akustischen Barrieren befinden. Die räumlichen Merkmale sollen dem Signal über eine Pegelabnahme nach dem $1/r$-Gesetz und einem überlagerten Dissipationsfilter, der die frequenzabhängige Luftabsorption nach Kapitel 2.3.1 abbildet, über die gesamte Distanz aufgeprägt werden. Darüber hinaus soll im Nahbereich ei-

ne Berücksichtigung der frequenzabhängigen Bodenreflexion nach Kapitel 4 auf Grundlage des Spiegelschallquellenmodells in Abbildung 4 stattfinden. Nachteilig an dem Modell der Schallstrahlapproximation ist, dass Beugungen und Interferenzen nicht Teil des Modells sind. [35, S. 8] Da die geometrische Akustik für flache Einfallswinkel zunehmend größere Fehler mit sich bringt [53, S. 60] und ohnehin für höhere Ordnungen sehr rechenintensiv wird, soll die Simulation der komplexen Reflexionen und Beugungen an Barrieren der Außenumgebung ab einer mittleren Entfernungen von $5 - 8$ m durch eine Überblendung in das mit einer prototypischen OIR gefaltete Audiosignale abgebildet werden.

4.2 Werkzeuge: Plattform, Hardware und Programmiersprachen

Die Entscheidung zwischen einer der weitverbreitetsten mobilen Plattformen Android und iOS ist auch die Wahl zwischen zwei Philosophien. Während Android ein offenes System ist, das auf vielen verschiedenen Geräten mit unterschiedlicher Hardware läuft, ist iOS proprietär und auf seine Hardware optimiert. Da diese Optimierung in dem Maß bei Android nicht möglich ist, fällt die Wahl zu Projektbeginn aufgrund der besseren Performance im Hinblick auf Audioanwendungen in Echtzeit [115] auf iOS. Erst kurz vor Projektende tauchen Nachrichten von einer ersten Low-Latency-App für Android [116] auf.
Aus der Entscheidung für eine native iOS-App ergibt sich *Objective-C* als primäre Programmiersprache mit *Xcode* als Entwicklungsumgebung (SDK). Objective-C ist eine objektorientierte Erweiterung der Programmiersprache C. Da C eine Teilmenge von Objective-C ist, kann jedes C-Programm mit einem Objective-C-Kompiler kompiliert werden. Für die Umsetzung der Audiosynthese wird die visuelle Open-Source-Programmiersprache *Pure Data* gewählt. Diese von Miller Puckette entwickelte Echtzeit-Audio-Programmiersprache ist in C geschrieben und kann über die von Brinkmann et al. [79] [80] entwickelte offene und freie *Programmierschnittstelle* (Application Programming Interface, API) *libpd* direkt in den App-Code eingebunden werden. Pure Data existiert in verschiedenen Versionen, so z. B. die von dessen Erfinder und Entwickler Puckette [81, 82] gepflegte Version (Vanilla) mit den grundlegenden Objekten zur seriellen Verarbeitung von Audiodaten und Zahlenströmen und die community-gepflegte Version PD-Extended[1] mit zahlreichen Externals. Bei diesen zusätzlichen Objekten handelt es sich wie bei den Basisobjekten um in C geschriebende Erweiterungen. Eine andere Möglichkeit der Einbindung externen Codes sind sog. Abstractions. Ein in Pure Data erstellter Objektzusammenhang, auch Patch genannt, kann als ein eigenständiges Objekt in Form einer Abstraction gespeichert und wiederverwendet werden. Die API libpd beinhaltet die grundlegenden Objekte der Vanilla-Version. Es steht dem Programmierer jedoch frei, diese mit externen oder selbst geschriebenen Objekten durch das Einbinden des jeweiligen Source-Codes zu erweitern. Ohne diese Möglichkeit wäre die Implementierung der DBS mit libpd nicht umsetzbar. Für Pure Data spricht in diesem Zusammenhang u.a die intuitive Erfassbarkeit des Datenflows auch für Außenstehende, im Falle einer Weiterentwicklung. Durch die Kommunikation über feste Kanäle mittels der libpd-Funktion `PdBase` sind App und Audiosynthese als Standalone separat entwickelbar. Desweiteren ermöglicht der modulare Charakter ein schnelles Austauschen und Testen von Teilen des Algorithmus'. Modularität wird oft mit Redundanzen und Overhead erkauft. Ob sich diese als ein Hindernis für die pd-basierte DBS auf dem Smart-

[1] Eine Übersicht zu Versionen und Forks findet sich auf der Webseite des IEM [117].

phone herausstellen, wird der Versuch zeigen. Die Existenz erfolgreicher Audio- und Spiele-Apps, die auf libpd zugreifen [118, 119, 120, 121, 122] unterstreicht jedoch das Potential der mobilen Audiosynthese mit libpd.
Als Hardware wurde das iPhone 5s als zum Zeitpunkt des Projektstarts aktuellstes iOS-Smartphone ausgewählt, welches mit dem ARMv8-basierten 64-Bit-Zweikernprozessor ca. 18.200 Millionen Instruktionen pro Sekunde (MIPS) [108] in realem Multithreading verarbeiten kann und sich damit bereits im Bereich von stationären Systemen bewegt [ebda.]. Da es sich trotzdem um ein Gerät mit geteilten Ressourcen durch mehrere gleichzeitig laufende Anwendungen handelt, steht natürlich nur ein Teil des Budgets für die Auralisation zur Verfügung.

4.3 Sensorbasiertes smartphoneinternes Head- und Positionstracking

Die zum Head-Tracking vorgesehenen Sensoren sind in iOS über Funktionen nativer iOS-Frameworks ansprechbar und sollen im Folgenden etwas näher betrachtet werden. Tabelle 2 zeigt die vorgesehene Verwendung der smartphoneinteren Sensoren.

Tabelle 2: Akquise der Positionsdaten

Sensor	Framework	Aufgabe
Unterstütztes Globales Ortsbestimmungssystem (Assisted Global Positioning System , A-GPS)	Core Location	Positionsbestimmung, Entfernungsberechnung, Adressenermittlung
Teslameter (Kompass)	Core Location/ Core Motion	Azimutdaten
Trägheitssensoren (Gyroskop, Accelerometer)	Core Motion	Elevationsdaten, Schritterkennung

Der Apple M7 Motion Co-Prozessor arbeitet als Sensor-Hub und nimmt alle Informationen von Kompass, Gyroskop und Beschleunigungssensor ununterbrochen entgegen und wertet diese mit Hilfe der Core Motion API von iOS aus.

A-GPS

Als *Globales Navigationssatellitensystem* (Global Navigation Satellite System, GNSS) bezeichnet man die Obergruppe satellitengestützter Navigationssysteme. Als US-amerikanisches *Globales Ortsbestimmungssystem* (Global Positioning System, GPS), europäisches *Galileo*, russisches *Globales satellitengestütztes Navigationssystem* (GLObal NAvigation Satellite System, GLONASS) und chinesischer *Großer Bär* (BeiDou, BDS) sind deren Untergruppen bekannt. Die Satelliten

des GPS senden in einem ungerichteten Radiosignal deren Zeit und Position. Aus den sich überschneidenden Kreisflächen kann der Standort des Empfängers berechnet werden. Der Empfänger benötigt zur Positionsbestimmung freie Sicht auf mindestens vier Satelliten: drei für die Laufzeitbestimmung und einen als Zeitreferenz für die Bestimmung des Zeitoffsets der Empfängeruhr [123]. Je mehr Satelliten in die Ortung einbezogen werden, desto genauer ist die Positionsbestimmung. Für die Nutzung in Mobiltelefonen wurde A-GPS entwickelt, das durch Datenabruf aus dem GSM-Netz die Ortung beschleunigen kann. Als *Globales System für Mobile Kommunikation* (Global System for Mobile Communications, GSM) bezeichnet man einen internationalen Mobilfunkstandard für digitale Netze. Die maximale horizontale Genauigkeit der zivilen GPS-Nutzung mit *Standard Ortsbestimmungsservice* (Standard Positioning Service, SPS) wird von der Air-Force mit $\pm 3,5$ m angegeben [124]. Hardwareseitig beherbergt der Qualcomm WTR1605L das GPS Empfangsmodul [125] mit GNSS-Unterstützung von GPS, GLONASS und BeiDou. Durch kombinierte Verwendung redundanter GNSS-Systeme ist eine höhere Zuverlässigkeit und Genauigkeit beabsichtigt.

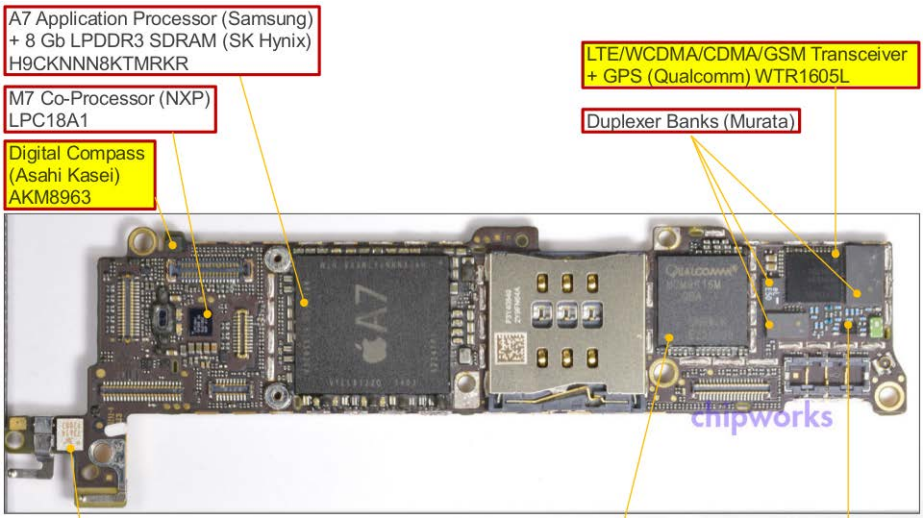


Abb. 9: Qualcomm WTR1605L/AKM AK8963, entnommen James [83, S. 7]

Teslameter

Der IMU Asahi Kasei Microdevices (AKM) AK8963 ist ein elektronischer 3-Achsen Kompass [125, 126], der eine Arbeits-, Rechen- und Verstärkereinheit miteinander kombiniert [125]. Bei einer Empfindlichkeit von $\approx 0,16\ \mu$T bei einer Wortbreite von 16 Bit oder $\approx 0,6\ \mu$T bei 14 Bit umfasst der Messbereich $\pm 4900\ \mu$T [126]. Das Initial-Offset bezeichnet die Abweichung des Sensors bei Null und beträgt bei diesem Modell in 14 Bit-Modus ± 500 LSB, was nach Gleichung 4.1 $6,25\%$ des verfügbaren Wertebereichs entspricht. Als *Niedrigwertigstes Bit* (Least Significant Bit, LSB) bezeichnet man die kleinstmögliche Auflösung eines Wertebereiches bei der Quantisierung.

Gleichung 4.1: niederwertigstes Bit (LSB) nach Lundberg [84]

$$T_Q = \frac{FSR}{2^N} = 1\ LSB$$

T_Q = kleinste Magnetfelddateneinheit (Quantum Tesla Level)
FSR = Wertebereich (Full Scale Range)
LSB = niederwertigstes Bit (Least Significant Bit)

Die Größe des LSB hängt von dem Verhältnis aus Messbereich und Wortbreite ab und ist äquivalent zu dem kleinsten unterscheidbaren Teil eines Wertebereichs. Die Datenupdates sind mit 8 Hz abrufbar [ebda.] und die Ausrichtung der X-, Y- und Z-Achse folgt dabei unter Verwendung der *„rechten-Hand-Regel“* der *Nord-Ost-Unten-Konvention* (North-East-Down-Convention, NED) [127].

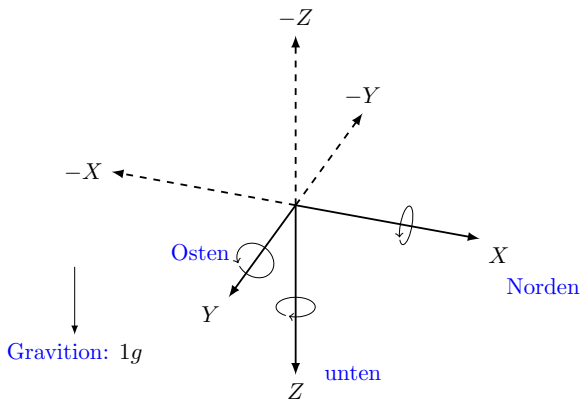


Abb. 10: Orientierung des Teslameters nach der Nord-Ost-Unten-Konvention

Die durch Schwankungen der elektrischen Ladung in der Ionosphäre hervorgerufenen Variationen der lokalen Stärke des Erdmagnetfeldes belaufen sich in einer Größenordnung von $0.25 \cdot 10^{-5}$ mT, was zu einer Variation von $0.03°$ [128] führt. Für den Einsatz des Teslameters zur Feststellung der Blickrichtung sind diese Schwankungen bezogen auf die menschliche Lokalisierungsunschärfe z. B. breitbandiger Schalle von $\pm 4°$ bei seitlichem Schalleinfall [14, S. 95] oder schmalbandiger Quellen zwischen $1 - 2$ kHz von $\pm 1°$ [15, S. 9] vernachlässigbar. Allerdings weist Apple darauf hin, dass metallische und magnetische Gegenstände in der Umgebung des Sensors einen starken Einfluss auf die Messwerte haben [85, S. 163] Ein einfacher Versuch ohne Einbeziehung der Z-Rotation [127], bei der das Smartphone in waagerechter Position um $360°$ um die durch den Sensor gehende Z-Achse rotiert, veranschaulicht in Abbildung 11 die Magnetfeldmessung. Die Messung wurde unter Verwendung von Matlab-Mobile und Matlab durchgeführt. Es wurden sowohl ferromagnetische als auch magnetische Gegenstände in die direkte Nähe des Gerätes gebracht. Ein

störungsfreier Durchlauf bildet einen Kreis ab – Unregelmäßigkeiten im Magnetfeld oder Sensordrift wirken sich verzerrend auf den Kreisradius und Mittel aus. Dass es trotz der Störversuche in mehr als 30 Durchläufen nicht gelang, typische Hard- oder Soft-Iron-Effekte [ebda.] auf Teslameter, wie ein Offset oder eine ellipsenartige Verzerrung zu rekonstruieren, legt das Vorhandensein komplexer softwareseitiger Korrekturalgorithmen seitens iOS nahe. Da die in den nativen Frameworks verwendeten Algorithmen nicht offenliegen, kann an dieser Stelle nicht näher auf die iOS-interne Lösung eingegangen werden. Jedoch finden sich detaillierte Informationen über Hard- und Soft-Iron-Effekte auf Teslameter und deren softwareseitige Kalibrierung, einschließlich mathematischen Erläuterungen der Zusammenhänge mit beigefügten Code-Bespielen bei anderen Herstellern wie z. B. Freescale Semiconductor [86] oder VectorNav Technologies [128]. Nichtsdestotrotz sollte in möglichst störungsfreier Umgebung eine Initial-Kalibrierung vorgenommen werden. Unter Umständen kann es je nach Umgebung vorteilhaft sein, die Kalibrierung auch während der Laufzeit zu wiederholen. Eine Möglichkeit dazu zeigt Listing 1.

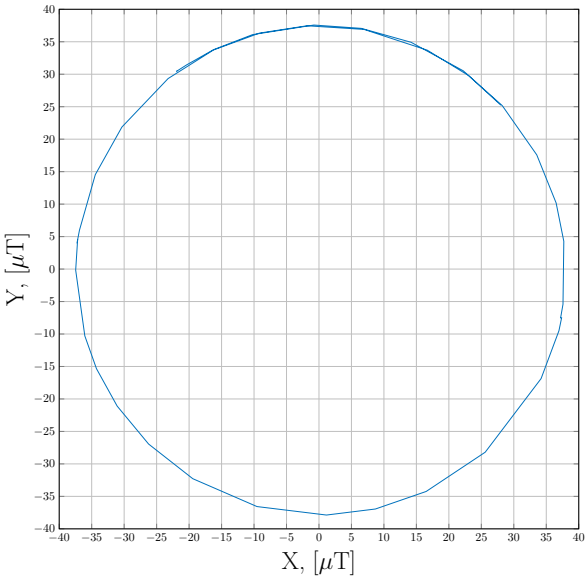


Abb. 11: Exemplarische Magnetfeldmessung unter Verwendung des Teslameters

Die Kumulation von Drift- oder Offsetfehlern eines einzelnen Sensors kann durch Einbeziehung weiterer IMUs in die Auswertung und durch die Entkoppelung der verwendeten Daten softwareseitig reduziert werden. Üblich ist z. B. die Kombination eines 3D-Accelerometers mit einem 3D-Gyroskop zu einer 6D-IMU. Durch die Nutzung von High-Level-Funktionen der nativen Frameworks und durch (initiale) Kalibrierung können bereits in iOS implementierte Lösungen genutzt werden.

Accelerometer

Auch ein Accelerometer ist ein IMU – als *Mikroelektromechanisches System* (Microelectromechanical System, MEMS) misst er die lineare Beschleunigung im Verhältnis zur Gravitationskraft $g = 9,81m/s^2$, welche oft als Vielfaches von g angegeben wird.

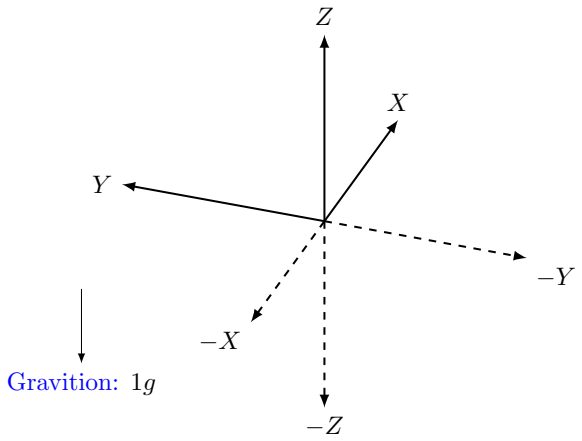


Abb. 12: Orientierung des Accelerometers

Die Ermittlung der Beschleunigung erfolgt mittels kapazitiver, resistiver, induktiver, piezoelektrischer, piezoresistiver oder optischer Effekte aufgrund der Masseträgheit, die auf eine seismische Masse in einem miniaturisierten Feder-Masse-System wirkt. Die Genauigkeit der Output-Daten resultiert aus dessen Varianz und Bias. Der prinzipielle Aufbau und die Funktionsweise sind ausführlich in Mineta et al. [87] beschrieben. Die vom Sensor gemessene Beschleunigung kann als Summe aus einem dynamischen und einem gravitativen Teil geschrieben werden [129].

Gleichung 4.2: Beschleunigungsvektor

$$a_{\text{Sensor}} = a_{\text{Dynamik}} + a_{\text{Gravitation}}$$

Ist über eine längere Zeitdauer der Betrag von $a_{\text{Sensor}} \approx 1g$, kann von einem Ruhezustand ausgegangen werden. Ändert sich der Wert dynamisch, wird die Beschleunigung durch Subtraktion von $|a_{\text{Gravitation}}|$ ermittelt, bis wiederum ein Ruhezustand eintritt.

Gleichung 4.3: Beschleunigungsmessung

$$|a_{\text{Sensor}}| = \sqrt{a_{\text{Sensor},X^2} + a_{\text{Sensor},Y^2} + a_{\text{Sensor},Z^2}}$$

Hier offenbart sich auch ein Mangel bei alleiniger Verwendung von Accelerometerdaten – erfolgt eine axiale Drehbewegung mit $|a_{\text{Sensor}}| \approx 1g$, würde ein Ruhezustand ausgegeben werden. Daher werden Accelerometer und Gyroskop meist kombiniert verwendet, was außerdem eine softwareseitige Fehlerkompensation ermöglicht. Gute Ergebnisse bei alleiniger Verwendung sind bei der Erfassung von Neigung und Orientierung in Ruhepositionen ohne dynamische Änderungen zu erzielen [ebda.]. Chipworks [125] identifizierten das Accelerometer des iPhone 5s als Bosch Sensortech BMA220.

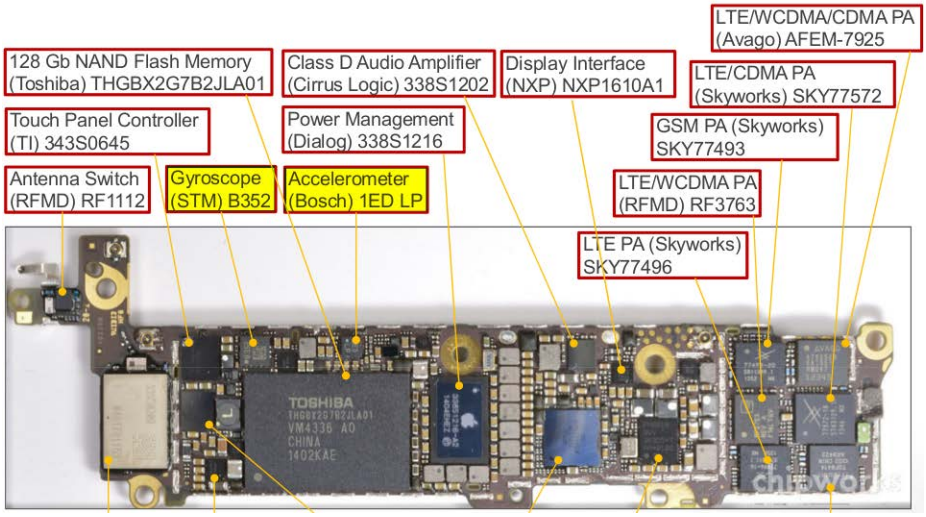


Abb. 13: Sensortech BMA220/STMicroelectronics B329, entnommen James [83, S. 8]

Dieses Bauteil weist einen typischen Bias von ± 95 mg [88, S. 6] auf, was lt. Chipworks zu einer Neigungsungenauigkeit von $\pm 5°$ führt. Der Wertebereich ist skalierbar von $\pm 2 - \pm 16$ LSB/g, dementsprechend variiert die Auflösung zwischen $62,5 - 500$ mg/LSB. Darüber hinaus lässt sich mittels skalierbarer Sleepzeiten von $2 - 200$ ms zwischen der Datenakquise u. a. der sensorbedingte Energieverbrauch beeinflussen [88, S. 10].

Wertebereich:	$\pm 2g$	$\pm 16g$
0000000000	0,061 mg	0,488 mg
0000000001	0,122 mg	0,977 mg
0000000010	0,183 mg	1,465 mg

Tabelle 3: Accelerometer BMA220

Gyroskop

Das im iPhone 5s erstmals verbaute 3x3mm Gyroskop L3GD20H von STMicroelectronics [125, 130] in Abbildung 13 ist ein 3-Achsen-Gyroskop und misst die Winkelgeschwindigkeit von Drehbewegungen unter Ausnutzung der Beibehaltung des Drehimpulses rotierender Systeme. Durch das bei einer externen Auslenkung von seiner Rotationsachse auftretende Drehmoment kann ein Gyroskop als Detektor für dynamische Rotationsbewegungen in Relation zu einer bekannten Initialposition eingesetzt werden.

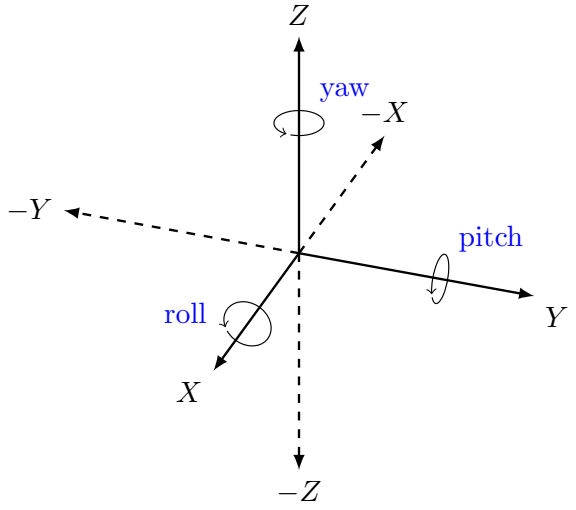


Abb. 14: Orientierung des Gyroskopes

Das L3GD20H besitzt bei einer Datenbitrate von 16 Bit einen skalierbaren Wertebereich von $\pm 245 - \pm 2000$ dps (degree per second), eine skalierbare Empfindlichkeit von $8,5 - 70$ mdps/LSB. Anscheinend sind in diesen Werte etwaige Toleranzen bereits inbegriffen, da sich bei einer rechnerischen Überprüfung für einen Wertebereich von ± 245 dps anders als im Datenblatt angegeben eine Empfindlichkeit von $\approx 7,48$ mdps/LSB ergibt. Weiterhin bietet der Sensor eine skalierbare Datenausgangsrate von $11,9 - 757,6$ Hz und ein Initialoffset von ± 25 dps[1] [131]. Aufgrund des Offsets und des Einflusses der Temperatur auf die Empfindlichkeit ist auch beim Gyroskop eine Kalibrierung nötig. Die Temperatur kann mit 1 Hz und 8 Bit Auflösung abgefragt werden. Der Hersteller schlägt in einem technischen Artikel [89] mathematische Methoden zur Datenauswertung und Sensorkalibrierung vor. In iOS erfolgt der Zugriff auf die Teslameterdaten entweder über Core Motion, bei dem man zwischen direkten und bereits vorverarbeiteten Daten wählen kann, oder über Core Location.

Core Location

Die Ortungsdienste werden in iOS durch die native API Core Location zur Verfügung gestellt. Das mit Hilfe der Klasse `CLLocationManager` erstellte Objekt `locationManager` ist die zentrale Instanz für Anfragen nach Positionsdaten. Dessen Properties sind u.a. die geografischen Koordinaten, die Genauigkeit der Datenpakte und die Update-Rate. Darüber hinaus ist es möglich, die Teslameterdaten auszulesen und über das einfache Setzen einer booleschen Property wie in Listing 1 Rahmenbedingungen für dessen Kalibrierung zu definieren. Durch die aufgerufene Kalibrierungsprozedur werden laut Apple Inc. [132] auch die übrigen IMUs kalibriert. Allerdings bleibt eine Restungenauigkeit, weshalb wie in Listing 1 der Maximalwert der zulässigen Ungenauigkeit nicht zu gering gewählt werden sollte.

[1] Dieser Wert gilt für einen gewählten Wertebereich von 2000 dps [89].

Listing 1: Core Location - Teslameter Kalibrierung

```
- (BOOL)locationManagerShouldDisplayHeadingCalibration:(CLLocationManager *)manager
{
    CLLocationDirection accuracy = manager.heading.headingAccuracy;
    return accuracy <= 0.0f || accuracy > 5.0f; // deviation in degree
}
```

Hinsichtlich der geografischen Ortung kann der Entwickler oder die Entwicklerin in Abwägung zwischen softwareseitiger Genauigkeit und Energieverbrauch eine der jeweilige Aufgabe gerecht werdenden Balance wählen. Die Schnittstelle bietet für die Ortung die Genauigkeitskonstanten aus Listing 2 an, welche als Wert der Property `desiredAccuracy` des Objektes `locationManger` gesetzt werden können und so dessen Arbeitsmodus bestimmen. Mit der Konstante `Best` erreicht man die maximale Genauigkeit von A-GPS/GLONASS[1], mit `BestForNavigation` werden zusätzlich zu den GSM- und GPS-Daten auch die IMUs abgefragt und ausgewertet. Für weitergehende Informationen zu den zahlreichen Funktionen des Core Location Framework sei auf dessen umfangreiche Reference [133] verwiesen.

Listing 2: Core Location - Genauigkeitskonstanten der Ortungsdienste

```
/* choose location accuracy:
*
*     kCLLocationAccuracyBestForNavigation  highest + sensor data
*     kCLLocationAccuracyBest               highest
*     kCLLocationAccuracyNearestTenMeters   10 meters
*     kCLLocationAccuracyHundredMeters      100 meters
*     kCLLocationAccuracyKilometer          1000 meters
*     kCLLocationAccuracyThreeKilometers    3000 meters
*/
locationManager.desiredAccuracy = kCLLocationAccuracyBestForNavigation;
```

Um einen Überblick über die mit dem iPhone 5s erreichbare Ortungsgenauigkeit zu bekommen, erfolgte eine Vergleichsmessung der Geokoordinaten mit Matlab Mobile[2] und Binaural Soundmap[3] am geodätischen Referenzpunkt Ludwig-Ehrhard-Ufer in Berlin Mitte [134, 135] unter freien Himmel und unter Feststellung der lokalen Empfangsbedingungen. Diese werden von einem *Garmin e trex 20* in Form der Zahl sichtbarer Satelliten erfasst. Der hochempfindliche GPS-Empfänger „sieht“ wie der Qualcomm WTR1605L im iPhone 5s sowohl GPS- als auch GLONASS-Satelliten und dient im Test als Vergleichsgerät.

[1] unter idealen Empfangsbedingungen
[2] App lief als Standalone, vgl. Abb. 15b
[3] Datenausgabe über Xcode Console, vgl. Anhang, Listing 9

Abb. 15: Ortungstest am geodätischen Referenzpunkt Berlin Mitte

(a) Garmin e trex 20

(b) iPhone 5s

Allgemein muss bei der Bestimmung der Distanz zwischen zwei geografischen Punkten hinsichtlich der sphärischen Verzerrung durch die Erdkrümmung eine Fallunterscheidung gemacht werden. Für große Distanzen ist die sphärischen Verzerrung unbedingt in die Berechnungen mit einzubeziehen. Für die sehr kleinen Distanzen in dieser Anwendung kann jedoch sind diese so gering, dass eine ebene Oberfläche angenommen und die Distanz nach Phytagoras berechnet werden kann.

Gleichung 4.4: Berechnung der Distanz zwischen zwei geografischen Koordinaten bei Annahme einer ebenen Oberfläche (gilt nur für kurze Distanzen)

$$s = \sqrt{sx^2 + sy^2}$$

$$sx = sl(\text{Longitude}_1 - \text{Longitude}_2)$$
$$sy = sb(\text{Latitude}_1 - \text{Latitude}_2)$$
$$sb = 111,3, \text{ Strecke zwischen zwei Breitenkreisen}$$
$$sl = sb \cdot cos\left(\frac{\text{Latitude}_1 + \text{Latitude}_2}{2} \cdot \frac{\pi}{180}\right), \text{ Strecke zwischen zwei Längenkreisen}$$
$$s = \text{Strecke zwischen den zwei geografischen Koordinaten}$$

Tabelle 4: Messwerte Ortungstest, sichtbare Satelliten: GPS: 11, GLONASS: 7

	Longitude	Latitude	Abweichung vom Referenzpunkt [Lon., Lat., m]
geodätischer Referenzpunkt	13,371688	52,522585	-
Garmin eTrex 20	13,37169	52,52261	$\Delta_{Lon} = 0,000098$, $\Delta_{Lat} = 0,001975$, $2,8m$
iPhone 5s *Matlab Mobile*	13,371636	52,522522	$\Delta_{Lon} = 0,000052$, $\Delta_{Lat} = 0,000063$, $7,8m$
iPhone 5s *Binaural Soundmap*	13,371683	52,522586	$\Delta_{Lon} = 0,000005$, $\Delta_{Lat} = -0,000001$, $3,5m$

Entgegen der von Matlab Mobile angezeigten horizontalen Genauigkeit von ± 5 m in Abbildung 15b errechnet sich aus den gemessenen Koordinaten mit Gleichung 4.4 eine Abweichung von $7,8$ m. Leider fanden sich keine Herstellerninformationen zu der in der Matlab-Mobile-App intern verwendeten Genauigkeitskonstanten. Ob die interne Konfiguration oder externe Veränderungen in den Empfangsbedingungen Grund für die Abweichung in der Ortungsgenauigkeit beider Apps waren, konnte hier nicht eruiert werden. In Binaural Soundmap wurde die maximale vom Gerät lieferbare Genauigkeit mit der Konstanten `kCLLocationAccuracyBestForNavigation` gewählt. Der Auszug der Ausgaben aus der Console zeigt im Anhang Listing 9 einen Mitschnitt der empfangenen Datenpakete, die diesen Angaben zugrundeliegen.

Core Motion

Das CMMotion-Framework ist in iOS die zentrale Schnittstelle für die Datenabfrage an die oben besprochenen IMUs. Analog zur Klasse `CLLocationManager`, wird für den Abruf der IMU-Daten eine Instanz der Klasse `CMMotionManager` initialisiert. Die Sensordaten können über die Schnittstellen direkt in blockweiser Verarbeitung abgerufen werden. Mit Blöcken ist es möglich, Code-Segmente als geschlossene Einheiten auszuführen, welche dabei auf Nebenthreads auslagerbar sind. Auch hier stehen die Sensordaten in verschiedenen Verarbeitungsstufen zur Verfügung. Diese beziehen sich in Core Motion einerseits auf direkte oder fehlerbereinigte Daten separater Sensoren, andererseits auf Bewegungseigenschaften wie Geräte-Orientierung oder Aktivität. Für letztere werden bereits die Daten verschiedener IMUs gemeinsam ausgewertet, was i. d. R. zu genaueren Ergebnissen führt. Daher ist es meist sinnvoll, sich als Programmierer nur dann von der High-Level-Ebene in Richtung Low-Level zu begeben, wenn in den High-Level-Klassen die gewünschte Funktionalität nicht gegeben ist. Einen umfassenden Überblick über die zahlreichen

Funktionen dieses Frameworks bietet die Core Motion Framework Reference [136].

Fazit für das Headtracking

Die Akquise der Headtracking-Daten erfolgt nach Möglichkeit auf High-Level-Ebene um die iOS-interne Fehlerkorrektur optimal zu nutzen. Während die Update-Raten der IMUs als geeignet bewertet werden, folgt aus den Eigenschaften des Teslameters die Notwendigkeit einer separaten Kalibrierung bei Anwendungsstart und eine Überwachung der gelieferten Daten während der Laufzeit. Als möglicherweise beeinträchtigend wird die Wetterabhängigkeit der geografischen Ortung eingeschätzt, da diese sich direkt auf die Berechnung der raumakustischen Parameter der Entfernung und Richtung auswirkt. Ungenauigkeiten in der Ortung wirken sich auf die Perzeption des Nutzers oder der Nutzerin als Verschiebung der virtuellen Klangquelle von der hinterlegten Position aus. Der prinzipielle Ablauf der akustischen Virtualisierung ist dadurch nicht betroffen. Eine wünschenswerte zukünftige allgemeine Verfügbarkeit von GDGPS mit einer horizontalen Genauigkeit von bis zu $0,1$ m [137] würde dieses Problem umgehend lösen. Positiv zu bewerten ist die gemessene horizontale Genauigkeit der Ortung mit dem iPhone 5s unter Verwendung der maximalen Genauigkeitskonstanten. Diese deckt sich mit offiziellen Angaben zur GPS-Nutzung mit SPS [124]. Weniger die konkrete Ungenauigkeit in der geografischen Ortung, als eine potentielle Unstetigkeit derselben wird als kritischer Punkt für die Umsetzung einer immersiven Auralisation innerhalb der MARA-Anwendung eingeschätzt.

4.4 *Binaural Soundmap* – Model-View-Controller

Wie von Apple empfohlen [138] richtet sich der Aufbau der App nach dem Entwurfsmuster *Modell-Präsentation-Steuerung* (Model View Controller, MVC). Dieses Architekturmuster für interaktive Systeme wurde 1979 von Reenskaug [139] für SmallTalk entwickelt und ist zu einem allgemeinen Standard anvanciert. Durch Trennung löst es das Problem einer oft schwer zu entwirrenden Verflechtung von Programm-Code und GUI. Das System nutzt ereignisbasierte *Rückruf-Funktionen (Call Back Functions)* (CBFs), die durch bestimmte Ereignisse ausgelöst werden. Das Model enthält die Datenstruktur, welche aufgrund ihrer Alleinstellung einfach für neue Anwendungen exportierbar ist. Sie ist völlig unabhängig vom View. Beide kommunizieren nur durch den Controller als Vermittler. Dieser steuert die Interaktion mit dem Benutzer oder der Benutzerin, den Aufruf des GUI sowie das Update der Datenbank. Ziel ist eine einfache Erweiterung des Codes und dessen Wiederverwendbarkeit zu erleichtern, Interaktivität und Spezialisierung auf dem Gebiet der Arbeitsteilung[1]. In den folgenden drei Abschnitten wird die Struktur und der Datenfluss in *Binaural*

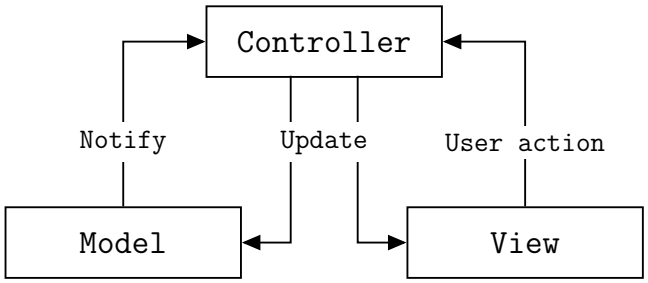


Abb. 16: Model-View-Controller (MVC)

[1] Frontend/Backend/Datenbank

Soundmap anhand des MVC-Entwurfmusters erläutert.

Model

Das Datenmodel wurde mit dem iOS-Framework Core Data entworfen. Dieses nutzt zur Datenspeicherung eine SQLite-Datebank. Hier wurden zwei Daten-Objekte *Soundwalk* und *Soundspot* als zentrale Entitäten dieser App mit einer One-To-Many-Beziehung (1:n) miteinander verknüpft. Im Soundwalk als ordnendes Element sind mehrere Soundspots kontextbezogen organisiert. Beiden Entitäten sind diverse Attribute zugeordnet. In den Attributen werden alle benötigten Eigenschaften von Soundwalk- und Soundspot-Objekten gespeichert. Aus den Entitäten werden Klassen generiert, mit denen sich später die Soundwalk- und Soundspot-Objekte als sog. `NSManagedObjects` erzeugen lassen. Das Präfix `NS` steht für die Klassenherkunft `NeXTStep` und ein `ManagedObject` ist ein Objekt, auf das nicht direkt, sondern nur über speziell dafür bestimmte und von Core Data bereitgestellte Funktionen zugegriffen werden kann. Die Attribute stellen für die Instanzen dieser Klasse die benötigten Properties. So beinhaltet ein Soundwalk neben Metadaten, wie Erstellungsdatum und Namen, eine grob nach Stadtbezirk aufgelöste Ortsangabe, den Kontext und eine Inhaltsbeschreibung. Der Soundspot beinhaltet die genauen geografischen Koordinaten, wenn vorhanden die postalische Adresse, die Ausdehnung des Geofences, den Audioinhalt, die akustische Umgebung in Form der jeweils hinterlegten OIR und Bodenoberfläche und weitere Metadaten wie Name und Erstellungsdatum als Property. Der Geofence bestimmt dabei die Ausdehnung des virtuellen Schallfeldes und dementsprechend die Entfernung, bei der die Audiosynthese freigeschaltet ist. Nähere Informationen zur Datenbankverwaltung mit Core Data findet sich im ausführlichen Core Data Programming Guide [140].

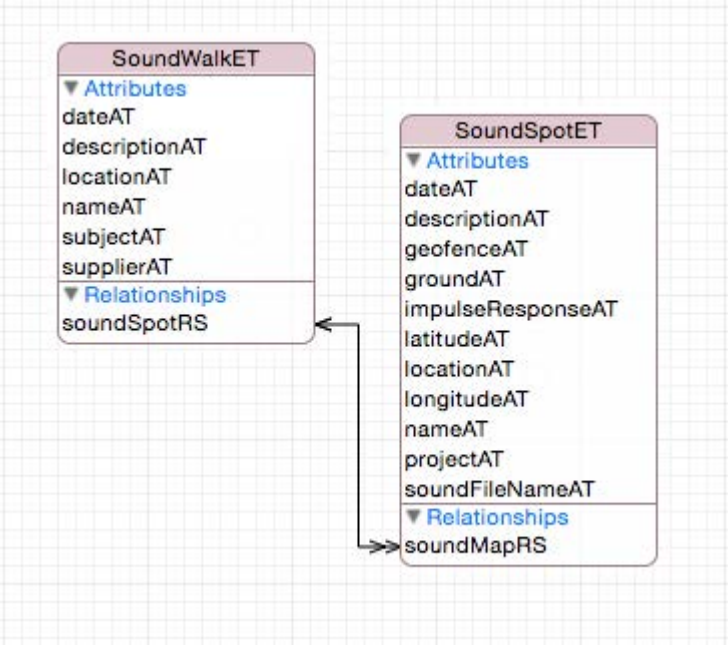
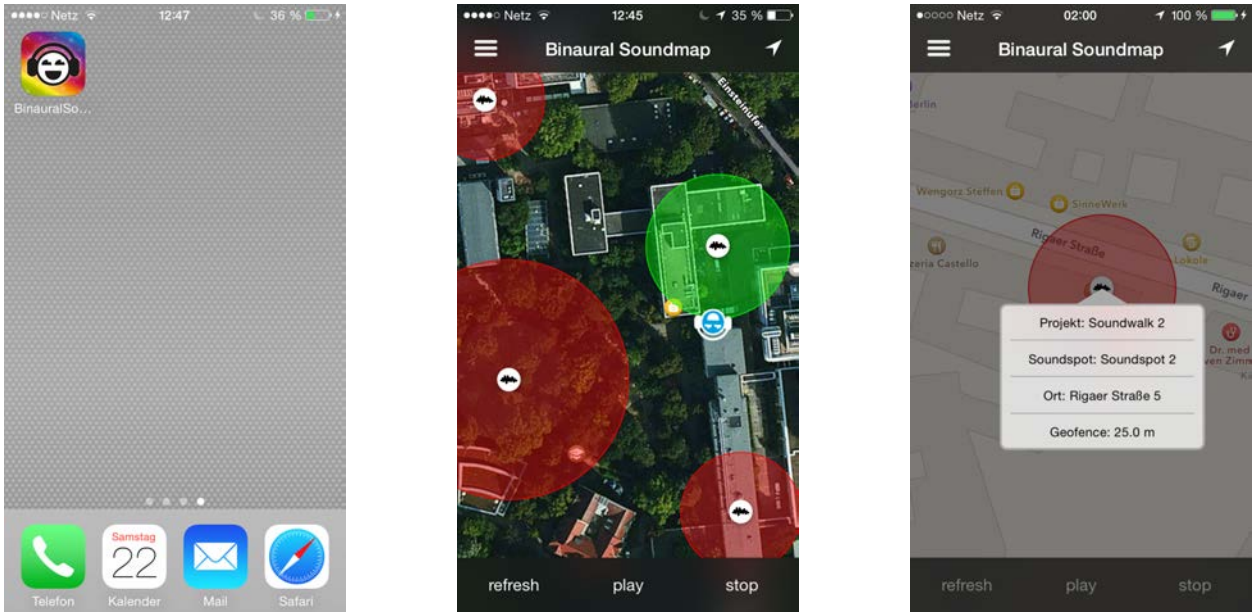


Abb. 17: Entitäten (ET) *Soundwalk* und *Soundspot* mit ihren Attributen (AT)

View

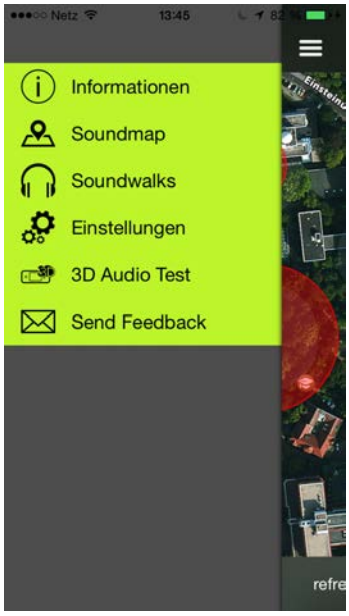
View-Objekte stellen die Daten im sichtbaren Teil des GUI zur Verfügung, dienen der Anzeige und Modifikation des Models und i. d. R. auch dem *Anwendererlebnis* (User Experience, UX). GUI-Elemente der App sind der `MapView` [141] für die Kartenansicht , der `UITableView` [142], der `UIPickerView` [143] als Auswahlmenü, der `UITextView` [144], `UISlider` [145], `UIButton` [146] und `UILabel` [147]. Nachdem die App über das App-Icon in Abbildung 18a gestartet wurde, öffnet sich der Begrüßungs-View, der mit einem kurzen Text über den Kontext und Zweck der App informiert. Von dort gelangt man direkt zur Kartenansicht in Abbildung 18b. Hier werden neben der geografischen Position der Soundspots durch animierte Soundspot-Annotationen auch weitere Model-Daten in Form von transparenten Overlays präsentiert. Diese Overlays zeigen mit ihrem Durchmesser die Ausdehnung des jeweiligen Geofences und durch Grünfärbung den aktuell nächstgelegenen und aktivierten Soundspot an, während die Übrigen rot unterlegt bleiben. Die Soundspot-Annotationen tragen ein Waveform-Logo und sind weiß pulsierend animiert. Durch eine Touchgeste auf die Soundspot-Annotation öffnet sich ein sog. Callout-View, der ausgewählte Metadaten des gewählten Soundspots anzeigt. Durch eine wiederholte Touchgeste wird die ausführliche Beschreibung angezeigt.[1] Ein sog. Long-Touch-Geste auf eine beliebige Position der Kartenansicht generiert einen neuen Soundspot. Dieser enthält vorerst Standarddaten- und -inhalte, welche in den Soundspot-Einstellungen in Abbildung 18g modifizierbar sind.

Abb. 18: Binaural Soundmap – GUI

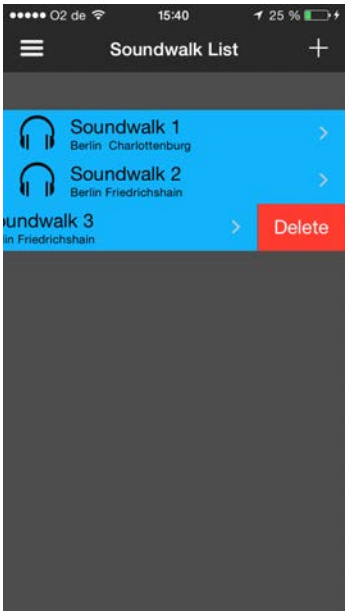


(a) App-Icon (b) MapView (c) Soundspot-Callout

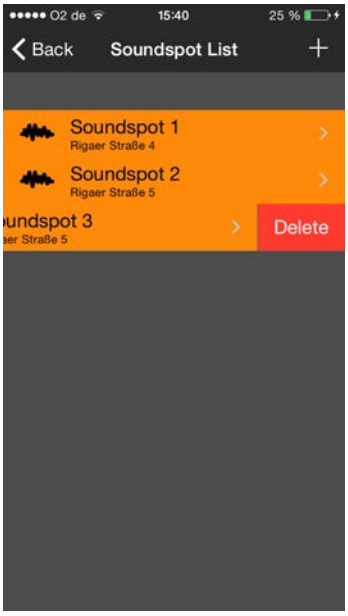
[1] Hier ohne Abbildung.



(d) Hauptmenü



(e) Soundwalk-Menü



(f) Soundspot-Menü

(g) Soundspot-Einstellungen

(h) Audiodaten-View

(i) Geodaten-View

Auch über das Soundspot-Menü in Abbildung 18f ist es möglich, neue Einträge zu erstellen. Wählt man hier das `Add-Icon` oben rechts im Navigation-View, wird ein Soundspot mit den aktuellen Geokoordinaten der Nutzer- oder Nutzerinnen-Position erstellt. Ein sich seitlich aufdeckender Reveal-View führt links zum Hauptmenü in Abbildung 18d. Von hier gelangt man in das Auswahlmenü der Soundwalks in Abbildung 18e, das wiederum auf die jeweilig zugeordneten Soundspots in Abbildung 18f und deren Einstellungen in Abbildung 18g verlinkt. In den

Soundspot-Einstellungen können neben den zugehörigen Metadaten, vor allem die Audio- und Geodaten angepasst werden. Durch Kombination von Audiodatei und OIR in der mittleren Zeile des zweispaltigen Audio-Picker-View in Abbildung 18h werden diese mit dem aktuell ausgewählten Soundspot verknüpft. Mit Hilfe des Schiebereglers in Abbildung 18i lässt sich der Durchmesser des kreisförmigen Geofences einstellen. Der Pickerview im unteren Bereich dient der Auswahl der prototypischen Bodenbedeckungen. Änderungen werden automatisch gespeichert. Weitere Menüpunkte in Abbildung 18d sind die App-Einstellungen, die aktuell nur die Wahl des Kartentyps beinhalten und ein Audio-Test-View mit dem die Auralisation abgekoppelt von der geografischen Ortung erlebbar ist. Dieser ist neben Testzwecken auch für eine eingeschränkte Demonstration innerhalb von Gebäuden nützlich. Über den Feedback-View ist eine Meinungsbekundung direkt an den Entwickler möglich. Das User-Icon in Abbildung 18b und zur größeren Ansicht in Abbildung 19 symbolisiert den aktuellen Standort und wird synchron zur Position von App-Nutzer oder -Nutzerin und Blickrichtung animiert.

Abb. 19: User-Icon

Controller

Der Controller gliedert sich in zwei übergeordente Einheiten: erstens den in Objective-C geschriebenen App-Controller und zweitens den in Pure Data implementierten und vom App-Controller gesteuerten Audio-Controller. Abbildung 20 veranschaulicht den systematischen Appaufbau und Signalfluss.

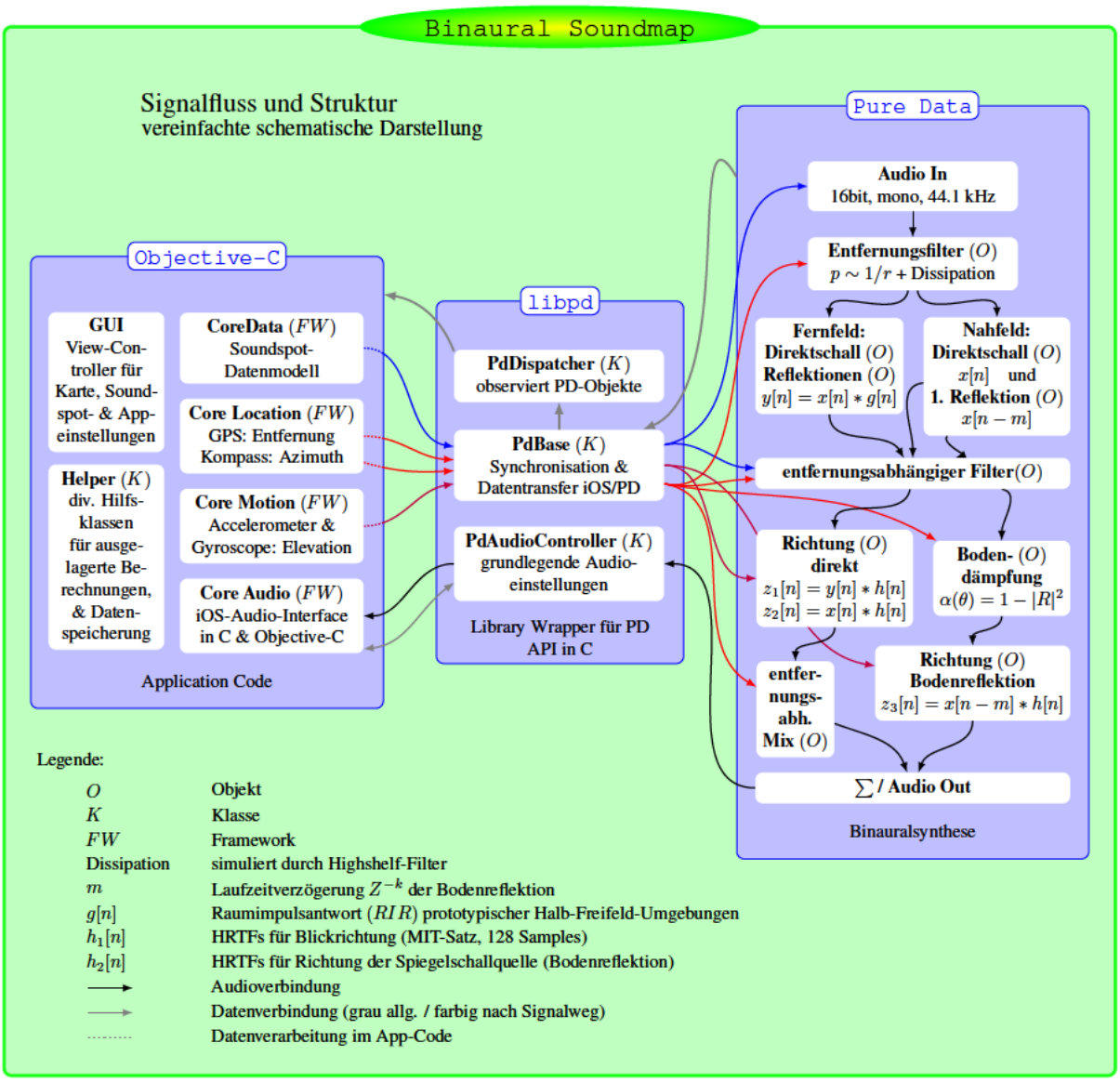


Abb. 20: wichtigste Komponenten und struktureller Aufbau von *Binaural Soundmap*

Der App-Controller steuert das GUI und übernimmt Aufgaben der Datenbankverwaltung, der Positionsbestimmung und der Bereitstellung der Steuerdaten für den die DBS vollziehenden Audio-Controller. Die drei blauen Blöcke symbolisieren das Zusammenspiel von nativer Programmierung in Objective-C und der Einbindung der Audio-Programmiersprache Pure-Data. Auch wenn beide die Programmiersprache C als gemeinsame Schnittmenge teilen, wird eine API für den Datenaustausch benötigt. Diese API stellt die von Brinkmann et al. [79] entwickelte offene Library `libpd` zur Verfügung. Während deren Klasse `PdDispatcher` PD-Objekte beobachtet und deren Daten an die App sendet, werden Daten in Form von Zahlen, Arrays, Strings oder Nachrichten durch die Klasse `PdBase` an den Pure-Data-Patch[1] geschickt. Die Klasse `PdAudioController` übernimmt für das Audio-Setup die Kommunikation mit dem Core-Audio-Framework. Die für DBS-Daten-Akquise zuständigen Elemente des App-Controllers im linken blauen Block und den die DBS vollziehenden Audio-Controller im Rechten werden in den nächsten Abschnitten eingehender betrachtet.
Nach dem MVC-Entwurfsmuster ist der Controller auch für die Aktualisierung des Models und die Übergabe dessen Daten an den View zuständig. In Abbildung 18e und 18f wird im GUI-Element `TableView` der aktuelle Zustand des `NSManagedObjectContext`, welcher den Objekt-Raum beschreibt [148], mittels `NSFetchedResultsController` [149] observiert. Dieser stellt die Ergebnisse der Datenbankabfrage dem GUI, hier dem `TableView`, zur Verfügung. Beides sind Instanzen von Core-Data-Klassen. Werden über das GUI Model-Objekte hin-

Listing 3: Core Data - NSFetchedResultsController

```
@property (strong, nonatomic) NSFetchedResultsController *fetchedResultsController;
@property (strong, nonatomic) NSManagedObjectContext *managedObjectContext;
@property (nonatomic, strong) SoundSpotET *soundSpotToDelete;

- (void)tableView:(UITableView *)tableView commitEditingStyle:(UITableViewCellEditingStyle)
    editingStyle forRowAtIndexPath:(NSIndexPath *)indexPath {
    if (editingStyle == UITableViewCellEditingStyleDelete) {
    // Delete object from database
    self.soundSpotToDelete = [self.fetchedResultsController   objectAtIndexPath:indexPath];
    [self.managedObjectContext deleteObject:self.soundSpotToDelete];

    if ([self.managedObjectContext hasChanges])
    [[DataStore sharedStore] saveManagedObjectContext:self.managedObjectContext];
    }

  if (editingStyle == UITableViewCellEditingStyleInsert) {

    // Add object to database
    SoundWalkET *soundSpotToAdd = [self.fetchedResultsController objectAtIndexPath:indexPath];
    [self.managedObjectContext insertObject:soundSpotToAdd];

  if ([self.managedObjectContext hasChanges])
      [[DataStore sharedStore] saveManagedObjectContext:self.managedObjectContext];
    }
}
```

zugefügt oder entfernt, aktualisiert der `NSFetchedResultsController` sowohl die Model-Ebene in Form der SQLite-Datenbank, als auch die View-Ebene alle Anzeige-Objekte des GUI

[1] Auch PD-Patch genannt stellt dieser einen Objektzusammenhang in Pure Data dar.

betreffend. Ein Beispiel für im Hintergrund ablaufende Prozesse auf Controller-Ebene ist die Ergänzung von ortsbezogenen Metadaten. Diese werden in Form der postalischen Adresse während der Erstellung eines Soundwalks- oder -spots über Reverse-Geocoding im Internet abgefragt und bei Verfügbarkeit als zusätzlich Information dem Model zugeführt und im GUI als Inhalt der Property `detailTextLabel` in der `UITableViewCell` [150] in Abbildung 18e/f zur Anzeige gebracht.

4.5 Dynamische Datenakquise und -verarbeitung

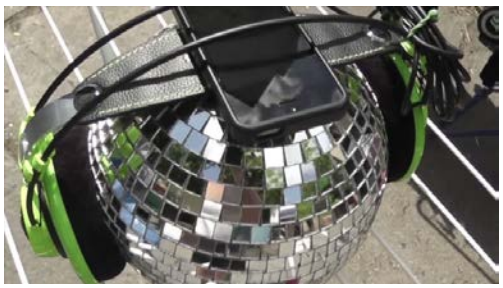


Abb. 21: Positions- und Head-Tracking

Die Generierung der Headtracking-Daten erfolgt in dieser Beta-Version wie festgelegt nicht durch eine externe Hardware, sondern wie in Abbildung 21 durch das Smartphone selbst. Zu diesem Zweck ist das Gerät auf dem Kopfhörerbügel befestigt und damit kann von den Gerätebewegungen auf die Bewegungen von Nutzer oder Nutzerin geschlossen werden. Die dafür genutzten Sensordaten werden in verschiedenen Verarbeitungsstufen durch die iOS-Frameworks Core Location und Core Motion zur Verfügung gestellt. Die zu erwartende Genauigkeit der von A-GPS und IMUs gelieferten Daten sowie die Art ihrer Bereitstellung wurde bereits in Abschnitt 4.3 dargelegt. Daher folgt in den nächsten zwei Abschnitten eine kurze Erläuterung der Gewinnung der für die DBS genutzten Steuerdaten auf App-Controller-Ebene.

4.5.1 Geodaten

Für die Generierung der Daten der räumlichen Parameter für die Distanz r und den Winkel φ der Azimutebene wird das Core Location Framework verwendet. Hierfür genügt ein zweidimensionales geografisches Modell, bei dem sich der User im Zentrum eines kartesischen Koordinatensystems befindet und die Position des Soundspots durch Betrag (r) und Winkel (φ) eines auf den Soundspot zeigenden Vektors beschrieben wird. Über die Abfrage der Property `horizontalAccuracy` und `timeStamp` können die empfangenen GPS-Datenpakete hinsichtlich ihrer Brauchbarkeit gefiltert werden. So werden Pakete mit einer größeren horizontalen Abweichung als ± 10m und einer Aktualität von > 10 s nicht als Steuerdaten verwendet. Dies beschränkt einerseits die App-Nutzung auf Situationen mit guten Empfangsbedingungen, verhindert jedoch andererseits unvorhersehbares Verhalten aufgrund unbrauchbarer Geodaten. Unter der Bedingung, dass sowohl ein Soundspot, als auch eine Position von Nutzer oder Nutzerin gefunden wurde, werden beide Koordinaten die Core-Location-Funktion `distanceFromLocation` übergeben und deren Ausgabe an Pure Data gesendet. Der Distanzfilter für die Positionsaktualisierungen wurde mit der Konstanten `kCLDistanceFilterNone` deaktiviert, so dass alle Bewegungen berichtet werden.
Ein Beispiel für eine CBF der Controller-Ebene ist der im `SoundmapVC` in Abbildung 18b arbeitende Schrittzähler der Klasse `CMPedometer` [151] des Core-Motion-Frameworks zur die Observierung der Bewegungsaktivität des Nutzers oder der Nutzerin. Befindet sich diese(r) außerhalb eines Geofences, werden nach einer festgelegten Anzahl Schritte die Abstandsberechnungen zum

Listing 4: Core Location - Distanzbestimmung

```
- (float)getDistanceFromSoundSpot {
    // need two positions
    if (self.currentUserPosition && self.currentSoundSpotPosition) {
    CLLocationDistance distance = [self.currentUserPosition distanceFromLocation: self.
        currentSoundSpotPosition];
    // prevent division through zero in PD and unpredicted behaviour
    if (distance >= 1) {
    distanceSendToPD = distance;
    } // To walk through the sound source is not implemented yet
    else distanceSendToPD = 1;
    // calculated distance will be sent to PD
    [PdBase sendFloat:distanceSendToPD toReceiver:@"distance_in_m"];
    return distance;
    }
    return 100; // if something is wrong - DBS won't start
}

- (void)viewDidLoad {
    [super viewDidLoad];
    // want all results from the location manager
    self.locationManager.distanceFilter = kCLDistanceFilterNone;
}
```

Distanz-Ranking der vorhandenen Soundspots ausgelöst. Auf diese Weise werden aufwändigere Berechnungen nur bei Bedarf und außerhalb einer Audioperformance ausgeführt. Zustandsvariablen dieser Art, die sich nur mit relativ niedriger Häufigkeit ändern, werden auf Controller-Ebene über Key Value Observing verwaltet. Beim Key Value Observing angemeldete Objekte werden beobachtet und Änderungen ihrer Properties automatisch als Nachricht an andere Objekte weitergeleitet, alternativ können CBFs aufgerufen werden.

4.5.2 Orientierungsdaten

Die Ausgabe der Geräteorientierung kann über das Core-Motion- oder das Core-Location-Framework geschehen. Bei der Nutzung der Teslameterdaten werden mit `trueHeading` vorzugsweise die

Listing 5: Core Motion - CMPedometer

```
@property (strong, nonatomic) CMPedometer *pedometer;
@property (nonatomic, strong) NSString *stepCounterString;
@property (nonatomic, strong) NSNumber *numberOfSteps;

- (void)startCountingSteps {
    self.numberOfSteps = 0;
    [self.pedometer startPedometerUpdatesFromDate:[NSDate date]
    withHandler:^(CMPedometerData *pedometerData, NSError *error) {
    dispatch_async(dispatch_get_main_queue(), ^{
    self.stepCounterString = [NSString stringWithFormat:@"Schritte: %@", pedometerData.
        numberOfSteps];
    self.soundMapView.userLocation.subtitle = self.stepCounterString;
    self.numberOfSteps = pedometerData.numberOfSteps;
    });
  }];
}
```

von ortsabhängigen magnetischen Abweichungen bereinigten Sensordaten genutzt. Das hat den Vorteil, direkt den in Listing 6 berechneten Winkel des Richtungsvektors von der User-Position zum Soundspot, für die Bestimmung des Azimutwinkels φ nutzen zu können. Bei einer Lokalisationsunschärfe in der Azimutebene mit $\pm 1°$ bis $\pm 4°$ [14, 15] ist es vertretbar, ganzzahlige Integerwerte als Steuerdaten der DBS zu verwenden.

Listing 6: Richtung des Soundspots bezüglich der User-Position

```
#define d2r ((M_PI)/180.0) // degree to radiant
#define r2d (180.0/M_PI)   // radiant to degree


- (void)getDirectionToCurrentSoundSpot {
  // calculate angle between user position and soundspot
    double diff_lat,diff_long,lat1,long1,lat2,long2,degree;
    lat1 = self.currentUserPosition.coordinate.latitude;
    long1 = self.currentUserPosition.coordinate.longitude;
    lat2 = [self.currentSoundSpot.latitudeAT doubleValue];
    long2 = [self.currentSoundSpot.longitudeAT doubleValue];
    diff_lat = (double) (((lat2*1000000)-(lat1*1000000))/1000000) * d2r;
    diff_long = (double) (((long2*1000000)-(long1*1000000))/1000000) * d2r;
    degree = (int)(r2d * (atan2(sin(diff_long) * cos(d2r*lat2), cos(d2r*lat1)*sin(d2r*lat2)-sin(
        d2r*lat1)*cos(d2r*lat2)*cos(diff_long))));

    if (degree >= 0) {
    self.currentSoundSpotDegree = degree;
    } else self.currentSoundSpotDegree = 360+degree;
}
```

Listing 7: Elevationsbestimmung

```
- (void)startUpdatingMotionData {
    NSOperationQueue *updateQueue = [[NSOperationQueue alloc] init];
    updateQueue.qualityOfService = NSOperationQueuePriorityHigh;
    [motionManager startDeviceMotionUpdatesToQueue:updateQueue
    withHandler:^(CMDeviceMotion *data, NSError *error) {
    int gravity = (int)(data.gravity.y * (-90));
    [self sendCurrentElevation:gravity];
  }];
}

- (void)sendCurrentElevation:(int)integer {
    if (elevation != integer) {
    [PdBase sendFloat:integer toReceiver:@"elevation"];
    elevation = integer;
    [self calculateThetaReflection];
  }
}
```

Der Elevationswinkel θ lässt sich direkt aus den Daten des Core Motion Framework berechnen. Mit Hilfe der Funktion `startDeviceMotionUpdatesToQueue: withHandler:` kann die vertikale Geräteorientierung über die Property `data.gravity.y`[1] ausgegeben werden. Die Lokalisationsunschärfe für die Elevation ist mit $\pm 4°$ bis $\pm 17°$ [90] verhältnismäßig grob, weshalb

[1] Vgl. Abb. 4.3

auch hier mit Integerwerten gearbeitet wird. Die Funktion `startDeviceMotionUpdates-ToQueue:withHandler:` übernimmt die blockweise Verarbeitung der gesammelten Sensordaten. Mit Hilfe der High-Level-Klasse `NSOperationQueue` [152] wird dieser Thread ausgelagert. Die Property `qualityOfService` bestimmt dabei die Priorität des Threads. Da es sich hier um die wichtige Generierung von Steuerdaten für die DBS handelt, wird diese durch die Festlegung auf `NSOperationQueuePriorityHigh` erhöht. Durch die parallele Verarbeitung der Sensordaten werden beide CPU-Kerne genutzt und die Ressourcen des Main-Threads bleiben für Eingaben über das GUI erhalten. Listing 8 zeigt die vorgegebenen Prioritätsstufen in der Objectice-C Deklaration der Property `qualityOfService` [ebda.].

Listing 8: NSOperationQueuePriority

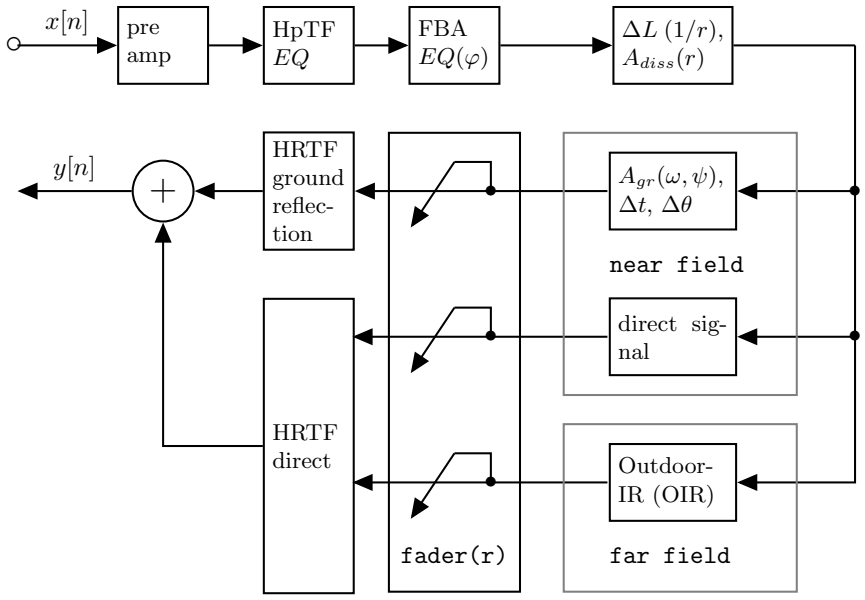
```
typedef enum : NSInteger {
  NSOperationQueuePriorityVeryLow = -8,
  NSOperationQueuePriorityLow = -4,
  NSOperationQueuePriorityNormal = 0,
  NSOperationQueuePriorityHigh = 4,
  NSOperationQueuePriorityVeryHigh = 8
} NSOperationQueuePriority;
```

Aus einem skalierbaren Wertebereich von $10 - 100$ Hz wurde die Updaterate unter Abwägung zwischen Echtzeitfähigkeit und Ressourcenverbrauch auf 25 Hz festgelegt.

4.6 Signaltechnische Simulation psychoakustischer und physikalischer Cues mit Pure Data

In den vorangegangenen Abschnitten wurden die für akustische Registrierung ausgewählten raum- und psychoakustischen Merkmale betrachtet. Abbildung 22 zeigt nun den entworfenen Signalfluss. Bei der Simulation der Distanz erfolgt eine Fallunterscheidung zwischen simuliertem Nah- und Fernfeld. Dabei wird im simulierten Nahfeld von einer ungehinderten Schallausbreitung im Halbfreifeld ausgegangen. Somit sind in erster Linie der Direktschall und die ihn überlagernde Bodenreflexion akustisch von Bedeutung. Hier kommt das Spiegelschallquellenmodel 1. Ordnung aus Abbildung 4 für kleine und mittlere Einfallswinkel ψ zur Anwendung. Die Dämpfung der auf dem Boden auftreffenden Schallwelle ist von der Schallkennimpedanz des Bodens sowie von der Frequenz und dem Einfallswinkel der auftreffenden Schallwelle abhängig. Die in Abbildung 5 dargestellten Absorptionsgrade prototypischer Bodenbedeckungen werden aus Gründen der Vereinfachung und Performanz mit Hilfe eines an diese Ergebnisse angepassten Polynomfilters mit dynamischen Koeffizienten approximiert. Mit größer werdendem Abstand verringert sich zum einem aufgrund des flacher werdenden Schalleinfallswinkels der Grad der Dämpfung [37] und zum anderen auch der zeitliche Versatz zwischen Direktschall und Bodenreflexion solange, bis beide nahezu zeitgleich das Ohr erreichen.

Abb. 22: Audiosynthese in Pure Data



$A_{gr}(\omega, \varphi) =$ winkel- und frequenzabhängige Bodendämpfung
$A_{diss}(r) =$ entfernungs- und frequenzabhängige Luftdämpfung
FBA $EQ_{\varphi} =$ winkelabhängiger Präsenzfilter

Außer diesem geringer werdenden Zeitversatz ist im Fernfeld eine große Zahl an Reflexionen und

Streuungen durch Bebauung und Vegetation akustisch wirksam, welche sich auf einem Smartphone nicht kostengünstig über Verfahren der geometrischen Akustik berechnen lässt. Daher erfolgt die Generierung zusätzlicher räumlicher Cues für das simulierte Fernfeld über eine Faltung des Eingangssignals mit prototypischen IRs, die sowohl frühe als auch späte Reflexionen enthält. Aufgrund der Verlagerung dieser Operation außerhalb des latenzkritischen Bereichs an den Anfang der Signalkette, kann die kostengünstigere Faltung im Frequenzbereich gewählt werden. Der latenzkritische Bereich beginnt mit dem Headtracking und schließt alle folgenden signaltechnischen Berechnungen mit ein, da die hier erzeugte Latenz bei einer Überschreitung zulässiger Grenzwerte direkt für den Nutzer oder die Nutzerin spürbar wird. Die Berechnungen des Audiosignals im simulierten Nah- und Fernfeld verlaufen zeitgleich, so dass zwischen beiden jederzeit verzögerungsfrei überblendet werden kann. Befindet sich der Nutzer oder die Nutzerin in einer Entfernung, die dem simulierten Hallradius entspricht, liegen beide Kanäle zu gleichem Anteil am Ausgang an.

4.6.1 Simulation der Schallausbreitung im Halbfreifeld durch Spiegelschallquellenmodell 1. Ordnung

Das bereits vorverstärkte und approximativ HpTF-entzerrte Eingangssignal wird nun entsprechend der berechneten Entfernung des Nutzers oder der Nutzerin vom Soundspot-Zentrum in Lautstärke und Klangfarbe moduliert. Die Anpassung der Lautstärke erfolgt über die Korrektur der Amplitude nach dem $1/r$-Gesetz.

Gleichung 4.5: Schallpegel in Abhängigkeit von der Entfernung zur Schallquelle

$$L_2 = L_1 - 20 \cdot log_{10} \cdot \frac{s_2[m]}{s_1[m]} \qquad [dB]$$

Die entfernungsabhängige Klangfarbenänderung aufgrund der frequenzabhängigen Luftschalldämpfung wird mit einem Highshelf-Filter approximiert, dessen Parameter sich an den ermittelten Werten in Unterkapitel 2.3.1 Abbildung 3 bei mittleren Temperaturen orientieren. Signaltechnisch wird dieser mit einem dynamisch angesteuerten Biquad-Filter nach Moser-Booth [153] umgesetzt. Dessen Koeffizienten werden über die Extension `highshelf` von Geiger [154] berechnet. Bei einem Biquad-Filter handelt es sich um einen rekursiven linearen Filter zweiter Ordnung mit Feed-Back und Feed-Forward-Gliedern. Gleichung 4.6 zeigt die allg. mathematische Darstellung eines solchen Polynomfilters.

Gleichung 4.6: Biquad Filter

normalisierte Differenzengleichung:

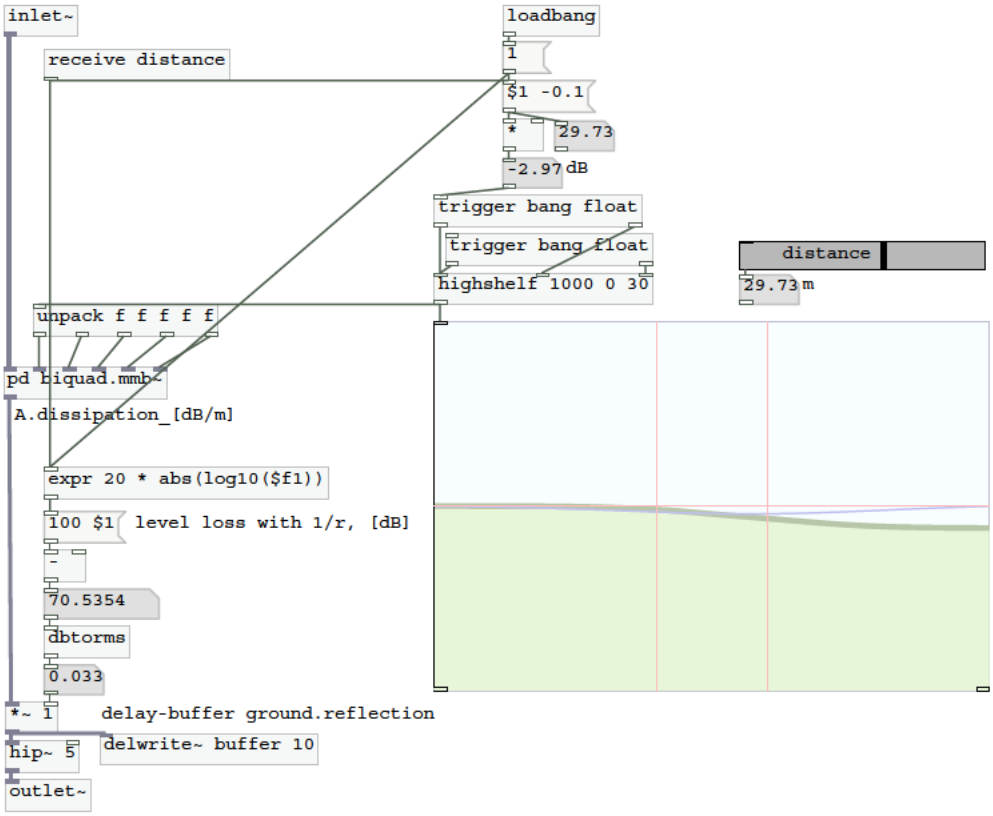
$$y[n] = b_0x[n] + b_1x[n-1] + b_2x[n-2] - a_1y[n-1] - a_2y[n-2]$$

mit Übertragungsfunktion:

$$H(e^{j\omega}) = \frac{b_0 + b_1e^{-j\omega} + b_2e^{-j\omega}}{1 + a_1e^{-j\omega} + a_2e^{-j\omega}}$$

Der Index gibt den Grad der Verzögerung an und der Koeffizient die Richtung. Die Berechnung der Koeffizienten des hier verwendeten Highshelf-Filters [154] folgt den Angaben von Robert Bristow-Johnson's Audio-EQ-Cookbook [155]. Dabei bestimmen die Koeffizienten a_x die zwei Polstellen und b_x die zwei Nullstellen der Übertragungsfunktion. Die Pure-Data-Abstraction von Moser-Booth [153] konstruiert den normalisierten Biquad-Filter aus jeweils zwei seriell geschalteten nicht-rekursiven komplexen Pol- und Nullstellenfilter-Objekten. Abbildung 23 zeigt die Implementierung in Pure Data.

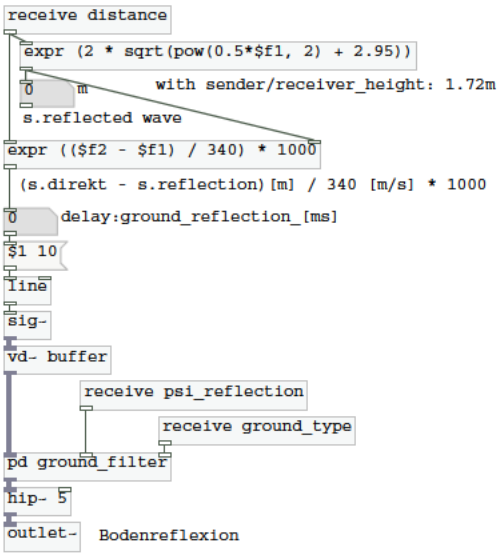
Abb. 23: geometrische und dissipationsbedingte Dämpfung



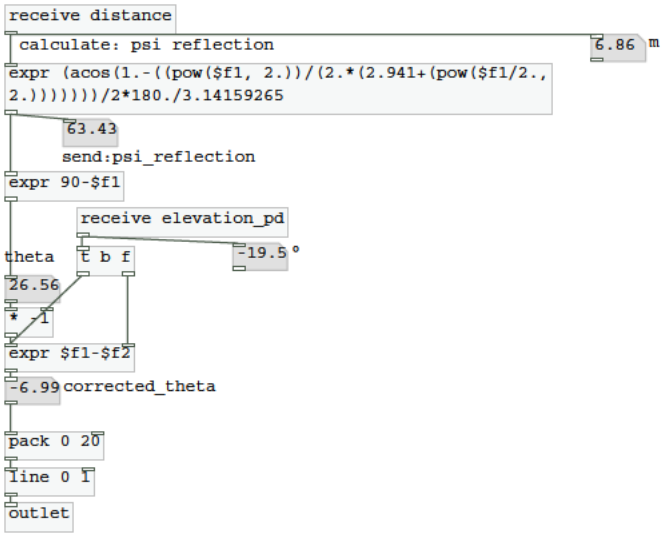
Am Ausgang dieses Subpatches enthält das Signal die geometrische und dissipationsbedingte

Dämpfung. An dieser Stelle erfolgt eine Zwischenspeicherung in einem Delay-Buffer, um das Signal für die Simulation der Bodenreflexion mit einer entsprechenden Laufzeitverzögerung zur Verfügung zu stellen. Die Berechnung der Delay-Zeit in Abbildung 24 erfolgt unter der vereinfachten Annahme gleicher Sender- und Empfängerhöhe für eine geschlechtsneutrale Durchschnittsgröße von 1, 72 m [156, 157].
Die einfallswinkel- und frequenzabhängige Bodendämpfung wird im PD-Subpatch `ground_filter` ebenfalls mittels dynamisch gesteuerten Biquad-Filter approximiert. Für die winkel- und frequenzabhängige Dämpfung der Bodenreflexion wurde im Subpatch `pd ground_filter` die bereits besprochene Approximation mittels eines Biquad-Filters durchgeführt, dessen Koeffizienten sich an den Berechnungsergebnissen aus Abschnitt 2.3.2 in Abbildung 2.18 orientieren.

Abb. 24: Elevation der Bodenreflexion



(a) Verzögerungszeit



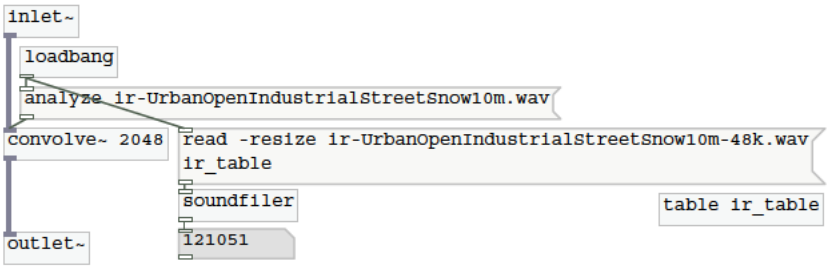
(b) Theta Bodenreflexion

4.6.2 Einfügen räumlicher Cues durch Faltung mit RIRs

Die Möglichkeit der drastischen Reduzierung der HRTF-Länge eröffnet den Weg eines gesplitteten Auralisations-Algorithmus', bei dem erst über eine latenzbehaftete aber günstigere Faltung mit einer monoauralen OIR im Frequenzbereich die gewünschten räumlichen Cues auf das Audiosignal übertragen werden um anschließend mit vertretbaren Kosten und ohne Anfangslatenz die Richtungsinformationen über eine Faltung mit entsprechend der Kopfposition ausgewählten HRTFs im Zeitbereich auf das bereits verhallte Audiosignal zu übertragen. Diese Vorgehensweise unterstützt die für eine Immersion essentielle simultane Rückkopplung von Kopfbewegung und Hörereignisempfindung und gibt zudem die Möglichkeit auf eine große Anzahl vorhandener OIRs zurückzugreifen.
Die Verhallung des Audiosignals findet im Audioalgorithmus wie in der analogen Welt vor der kopf- und körpergeometriebedingten Klangmodulation statt. Die Anwendung nutzt zu diesem

Zweck das Pure-Data-External `convolve~` von Brent [158], mit dem eine Partionierte Faltung[1] im Frequenzbereich unter Verwendung der komplexen FFT und dem OLA-Verfahren durchgeführt wird. Die Latenz und Prozessorlast kann über die wählbare Blockgröße auf ein tragfähiges Maß angepasst werden.

Abb. 25: Verhallung durch Faltung mit OIR



Die Impulsantworten entstammen der Sammlung Outdoor-Impulsantworten von Boom Library [159]. Der große Vorteil dieser Sammlung ist die genaue Dokumentation der Aufnahmeumgebung. Ein jeder Aufnahme zugeordnetes Foto des Ortes gibt Auskunft über die Art der Bodenbedeckung, wie Schnee, Gras, Sand oder Asphalt. Darüber hinaus ist zu jeder Aufnahme der Abstand zwischen Impulsquelle und Mikrofon vermerkt. Dieser legt das Verhältnis von Direkt- und Diffusschall starr fest. In der Implementierung werden die zwei Kanäle des Nah- und Fernfeldsignals entfernungsabhängig gemischt, so dass bei der Auswahl der OIR die maximale Ausdehnung eines Soundspot-Geofences zu berücksichtigen ist.

4.6.3 Einfügen binauraler Cues durch Faltung mit HRTFs

Mit dem PD-Extended-Objekt `earplug~` von Xiang et al. [23] steht ein passendes Faltungstool zur Verfügung. Es nutzt ein gerastertes Set von HRTFs, das auf den Kemar-Messungen Gardners von 1994 [114] mit Modell DB-4004 basiert, welche im reflexionsfreien Raum des MIT stattfanden [91]. Die Messung erfolgte am geblockten Ohr in einem Abstand von $1,4$ m mit einer angenommenen Abweichung von $> 1,5$ cm, was einer Winkelgenauigkeit von $\pm 0,5°$ entspricht [ebda.]. Als Testsignal wurde eine *Maximallängenfolge* (Maximal Length Sequence, MLS) verwendet. Das Pure-Data-Objekt `earplug~` bildet durch lineare Interpolation zwischen den 368 HRTFs volle 360° in Azimut- und 130°[2] in Elevationsrichtung ab. Für die Interpolation werden die vier Stützpunkte I_a, I_b, I_c, I_d verwendet, die den interessierenden Punkt I_P einschließen. Bedingt durch das HRTF-Aufnahmeverfahren [91] befinden sich jeweils zwei davon auf derselben Horizontalebene, d. h. I_a, I_b und I_c, I_d besitzen dadurch dieselben Latitude-Koordinaten. Gleichung 4.8 zeigt das in `earplug~` angewendete Verfahren, wobei Y die Latitude- und X die Longitude-Koordinaten darstellen.

[1] Vgl. Abschnitt 3.1.2
[2] $-40°$ bis $+90°$

Gleichung 4.7: HRTF-Interpolation nach Xiang et al. [23], Schritt 1

$$I_P = \frac{Y_P - Y_{ab}}{Y_{cd} - Y_{ab}} \cdot \left(\frac{X_P - X_a}{X_b - X_a} \cdot I_a - \frac{X_b - X_P}{X_b - X_a} \cdot I_b \right) + \frac{Y_{cd} - Y_P}{Y_{cd} - Y_{ab}} \cdot \left(\frac{X_P - X_c}{X_d - X_c} \cdot I_c - \frac{X_d - X_P}{X_d - X_c} \cdot I_d \right)$$

Auf diese Weise kann für jeden beliebigen Punkt im Wertebereich eine HRTF errechnet werden. Der Übergang von einem Punkt zum nächsten erfolgt wiederum mittels linearer Interpolation zwischen dem aktuellen und dem vorangegangenen Wert.

Gleichung 4.8: HRTF-Interpolation nach Xiang et al. [23], Schritt 2

$$y(k) = \sum_{n=0}^{127} x(k-n) \cdot \left(\frac{k}{\text{blocksize}} \cdot I_P(n) + \frac{\text{blocksize} - k}{\text{blocksize}} \cdot I_{P_0}(n) \right)$$

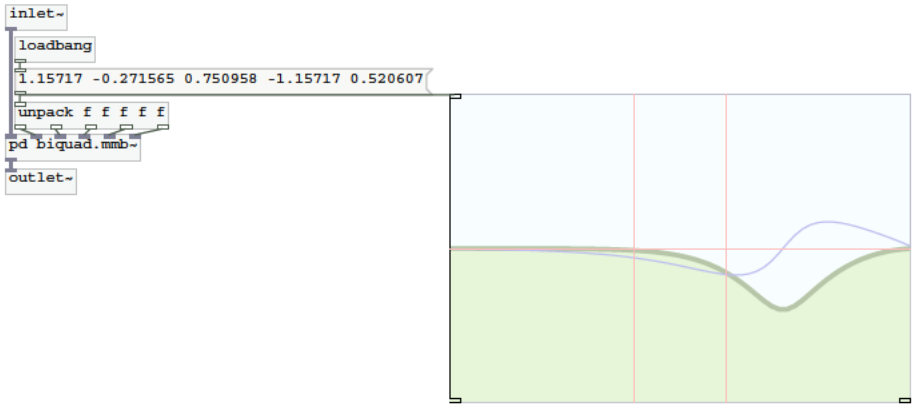
$$\begin{aligned} k &= 0, 1, \cdots, \text{ blocksize} - 1 \\ I_{P_0} &= \text{IR des vorangegangenen Blocks} \\ I_P &= \text{IR des aktuellen Blocks} \end{aligned}$$

Das verwendete HRTF-Set liegt im Textformat vor und ist somit gegen neuere oder individuelle HRTFs austauschbar. Von besonderer Bedeutung für die mobile Anwendbarkeit der hier verwendeten direkten Faltung ist die Länge der IRs, die basierend auf den Ergebnissen von Hörversuchen der Entwickler Xiang et al. [23] unter Erhalt der Lokalisationsleistung auf 128 Samples gekürzt wurden.

4.6.4 Verbesserung der binauralen Abbildung durch approximierte HpTF-Kompensation

In Møller et al. [73, S. 224, Abb. 3b] ist eine Anhebung in diesem Bereich sowohl für das geblockte als auch für das offene Ohr zu sehen wobei sich beide ab 2 kHz innerhalb lokaler Peaks voneinander unterscheiden. Die gewählte Filterkurve des Band-Shelf-Filters orientiert sich entsprechend der in dieser Arbeit verwendeten HRTFs an die Zielfunktion für das geblockte Ohr und geht von der Verwendung circumauraler HP aus. Um für verschiedene HP-Abstimmungen eine Linearisierung für diesen Bereich zu erzielen, wurde eine geringe bis mäßige Flankensteilheit gewählt. Der Parameter Gain von -10 dB orientiert sich dabei an der Zielfunktion für diffusfeldabgestimmte HP [73, S. 225, Abb. 5b].

Abb. 26: Approximierte HpTF-Kompensation in Pure Data, $f_m = 3$ kHz, Gain= -10 dB

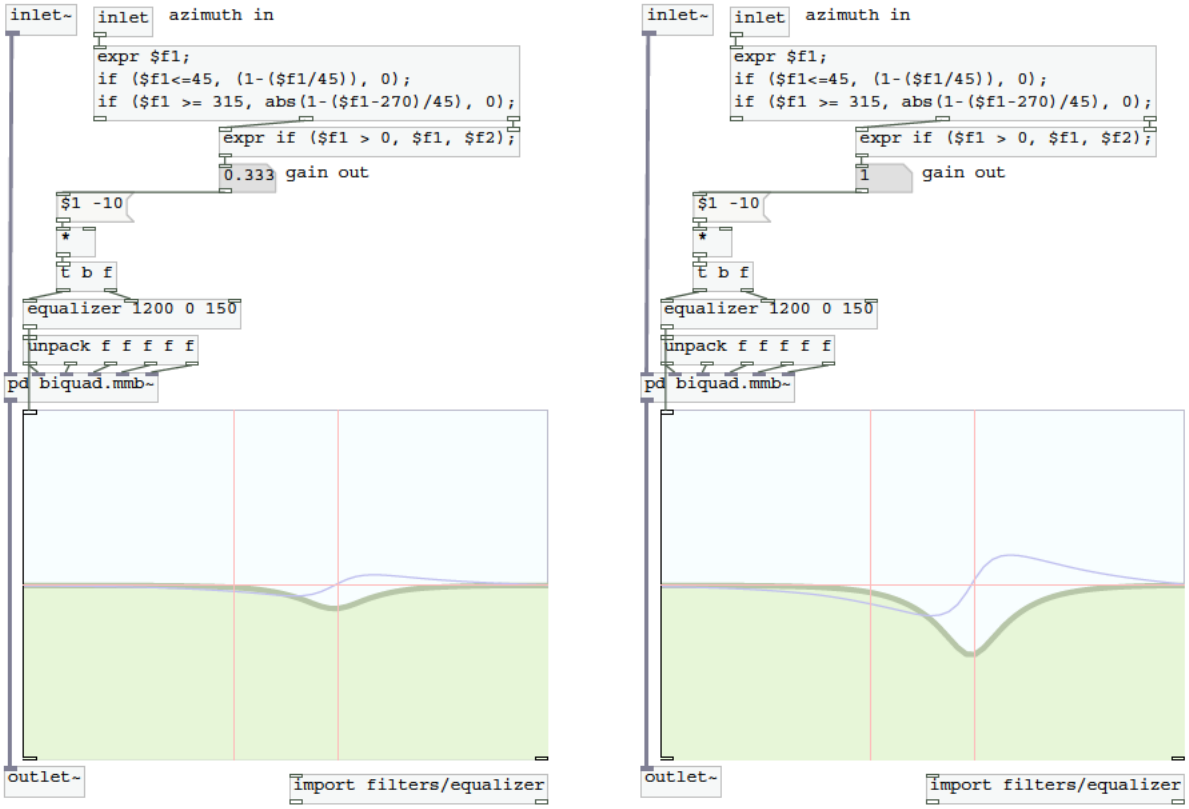


In Pure Data kommt zur Erstellung eines Band-Shelf-Filters die in Abschnitt 4.5 erwähnte Biquad-Abstraction von Moser-Booth [153] zum Einsatz, dessen Parameter in diesem Fall jedoch statischer Natur sind und daher ressourcenschonend als Listen-Objekt hinterlegt werden können. Berechnet wurden die Biquad-Koeffizienten mit Hilfe des PD-Externals `equalizer~` von Geiger [160] nach Bristow-Johnson [155].

4.6.5 Enhancements

Der Effekt richtungsbestimmender Bänder auf die Lokalisation soll hier in einem sounddesignerischen Enhancement nutzbringend eingesetzt werden. Als theoretischer Hintergrund und Datenqelle dient hier ein Experiment von Blauert und Braasch [14, S.94], bei dem die Hörereignisrichtung als Funktion der Mittenfrequenz terzbandiger Rauschsignale bestimmt wurde. Hier fällt ein Bereich um $1,2$ kHz auf, bei dem sich die richtungsbestimmenden Bänder der Vorne- und Hintenlokalisation als Pole gegenüberstehen, ohne dabei von für die Oben-Lokalisation wichtigen Frequenzbändern überlagert zu sein, wie das im hohen Frequenzbereich der Fall ist. Dabei unterstützt ein Peak mit dieser Mittenfrequenz die Hinten- und ein Notch die Vorne-Lokalisation. Gerade diese gegensätzliche Wirkung dieses Frequenzbandes macht es interessant für eine Effektverstärkung. Aufgrund der bereits im Zusammenhang mit der Hörbarkeit von Kammfiltereffekten erwähnten besseren Hörbarkeit von Peaks gegenüber den Notches [32, 92] wird ein richtungsselektiver Notchfilter zur Unterstützung der Vorne-Lokalisation in die Signalkette eingefügt.

Abb. 27: winkelabhängiger EQ bei $f_m = 1,2$ kHz zur Minderung der FBA



(a) bei $\varphi = 30°$ (b) bei $\varphi = 0°$

Der Blinkwinkel bestimmt hier die Dämpfung in einem Wirkbereich von $-45°$ bis $+45°$und erreicht bei $0°$ sein Maximum bei experimentell gewählten -10 dB. Das Ergebnis ist eine zunehmende Präsenz hin zur Richtung um $0°$ Azimut, was einer Vorne-Lokalisation zugute kommen soll. Subjektiv tritt eine spürbare Verbesserung der Unterscheidbarkeit zwischen frontaler und rückwärtiger Beschallung ein. Die richtungsbestimmenden Bänder sind individualspezifisch durch die Pinnaform und -größe bestimmt und bereits in den HRTFs enthalten. Nach Untersuchungen von Burkhard [93] führte die Verwendung nicht-individueller HRTFs, die von Personen mit großen, seitlich hervorragenden Ohren stammten, zu einer deutlichen Reduzierung der FBA auch bei anderen Personen. Dies wies in die Richtung, dass die Verstärkung vorhandener, bekannter psychoakustischer Effekte zu besseren Ergebnissen bei der Lokalisation führen kann. Unter anderem ist der „Satz große Ohren“ für den Kemar-Dummy-Head ein Resultat dieser Untersuchungen von Maxwell und Burkhard [94].

5 Technische Leistungsbetrachtung

Um die Audioperformance innerhalb der App möglichst separiert beurteilen zu können, verzichtet der Audio-Test-View mit seinem reduziertem GUI auf grafische Animationen und Distanzberechnungen. Lediglich Sensoren wie die IMUs und deren nachgeschaltete Funktionen für die Azimut- und Elevationsbestimmung sind dort aktiv. Zu Messzwecken wurden alle für die iOS-App nicht benötigten PD-Objekte entfernt. Dazu gehören die zu Demonstrationszwecken in den verwendeten Abbildungen eingefügten Nummern-Objekte, Filtervisualisierungen und Fader. Der benötigte RAM beläuft sich im Audio-Test-View auf ≈ 40 MB. Die Auslastung der Doppelkerne bewegt sich zwischen $20\% - 30\%$ und liegt damit in einem optimalen Bereich.
Im Mapview ist der Ressourcenbedarf etwas höher. Abgesehen von den grafischen Soundspot-Animationen in der Kartendarstellung finden hier zusätzlich noch etliche Schritte der Ortung und Entfernungberechnung statt, die im Audio-Test-View teilweise durch einfache Schieberegler ersetzt sind. Der Speicherbedarf des RAM beträgt hier ≈ 80 MB. Die Prozessorlast schwankt zwischen $40\% - 50\%$ und weist sporadische Peaks bis 70% auf.

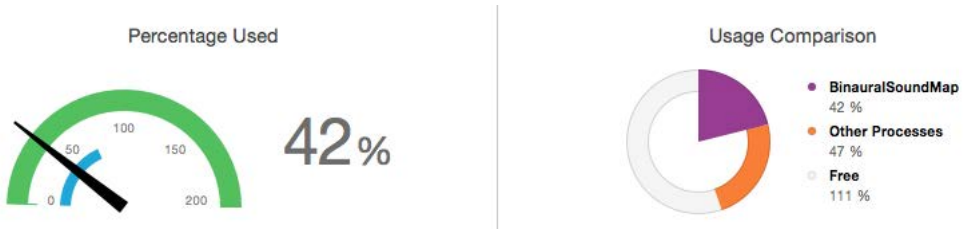


Abb. 28: CPU-Nutzung im Vollbetrieb

Diese Prozentangabe bezieht sich dabei auf einen der zwei vorhandenen Kerne, d.h. bei einer Last $< 100\%$ befindet sich ein Kern im Leerlauf. Die Audiosynthese verläuft dabei ungestört und unterbrechungsfrei. Bei einer Abkopplung der Audiosynthese vom Mapview wurden für Grafik, Ortung und Entfernungsberechnungen nur $5\% - 7\%$ der verfügbaren CPU-Leistung beansprucht.
Der Unterschied zwischen dem Ressourcenverbrauch der beiden Programmsektionen im abgekoppelten und zusammengeschalteten Modus weist auf Optimierungsreserven bei der Kommunikation zwischen beiden Programmteilen hin. Generell gilt, dass sich Pure-Data-Objekte hinsichtlich ihres Ressourcenbedarfs und somit in ihrer Eignung für mobile Apps stark unterscheiden.
Wie erwartet und durch das Profiling bestätigt, erweist sich `earplug~` aufgrund der direkten Faltung als das CPU-intensivste PD-Objekt. An dieser Stelle ist anzumerken, dass `earplug~` nicht für den mobilen Einsatz geschaffen wurde und dessen Entwickler Xiang et al. [23] vom Einsatz mehrerer dieser Objekte auf einem herkömmlichen Computer (2005) absahen. Aufgrund der Leistungsfähigkeit der verwendeten Hardware konnten hier dennoch zwei Richtungen in 3D mit zwei Objekten diesen Typs simultan abgebildet werden. Auf Platz zwei stehen in der Summe die vier verwendeten rekursiven IIR-Filter, die als zeitkritische Objekte ebenfalls in der CPU-intensiven Direktform implementiert sind. Das häufig verwendete `expr`-Objekt zur Berechnung komplexer mathematischer Formeln fällt als drittes Objekt mit deutlicher CPU-Intensität auf.
Der Störanfälligkeit des Teslameters durch äußere Einflüsse wurde mit einer initialen Kalibrierung und fortlaufenden Überwachung der gelieferten Daten begegnet. Hinsichtlich der Grenzen

der geografischen Ortungsgenauigkeit mittels A-GPS unter SPS kristallisieren sich die Unstetigkeiten in der Ortung als ein gravierenderes Problem für die MARA-Anwendung heraus, als die Positionsabweichung selbst. Unter optimalen Empfangsbedingungen sind bei einer horizontalen Genauigkeit[1] von ± 5 m Fehler bis zu 10 m möglich. Ein Positionssprung um wenige Meter in großer Entfernung zur virtuellen Schallquelle hat verhältnismäßig geringe Auswirkungen auf die Richtungssimulation, wenn sich die dadurch hervorgerufene Richtungsänderung im Rahmen der natürlichen Lokalisationsunschärfe bewegt. Im ungünstigsten Fall jedoch führt ein solcher Positionssprung in unmittelbarer Nähe eines Soundspots zu einer plötzlichen Richtungsänderung des wahrgenommenen Schallereignisortes. Subjektiv konnte dieser mit der Entfernung abnehmende Effekt in einer Distanz von $0 - 5$ m festgestellt werden. Der Effekt nahm ab, wenn sich das Gerät in Bewegung befand, da dadurch die Ortung unterstützt wurde.
Für die mögliche Latenz des Headtrackings, unterhalb welcher keine Beeinträchtigung der Immersion zu erwarten ist, werden in der Literatur Werte wie $\Delta t < 96$ ms bei Sandvad [20, S. 9] oder $\Delta t < 85$ ms bei Mackensen [95, S. 45] genannt. Die Updaterate von 25 Hz der für die Berechnung der Elevation genutzten Pitch-Daten aus Core Motion bestimmt unverändert die Latenz der Elevationsberechnung mit gemessenen 40 ms. Für die Azimutbestimmung liegt dieser Wert bei 20 ms (Anhang, Listing 10). An dieser Stelle zahlen sich die Kosten der direkten Faltung mit `earplug~` aus – für die Interpolation und Faltung einer 128 Samples langen HRTF mit einem entsprechendem Block des Audiosignals benötigt es nur $\approx 0,5$ ms (Anhang, Listing 13). Bei einer Samplerate von 48000 kHz sind das $< 20\%$ der systembedingten Latenz von $2,66$ ms aufgrund der Blockgröße von 128 Samples. Das Objekt `convolve~` benötigt für die Faltung mittels (I)FFT für einen 2048 Samples langen Block etwa $1 - 3$ ms (Anhang, Listing 14). Um genaue Zeitwerte zu erhalten, wurde der Durchlauf einer blockweisen Faltung durch Messung über die Systemuhr per Console-Ausgabe in der entsprechenden `*_perform`-Funktion gemessen. Diese Funktion ist Bestandteil eines jeden Pure-Data-Objektes und beinhaltet nur dessen Algorithmus. Die systembedingte Latenz beträgt aufgrund der Blockgröße beim `convolve~`- Objekt $42,66$ ms. Damit sind die Latenzen der Faltungsobjekte erwartungsgemäß hauptsächlich durch die gewählte Blockgröße determiniert. Für Binaural Soundmap konnte auf dem mobilen Gerät eine Gesamtlatenz vom Start der Audiosynthese bis zum ersten Output-Sample von durchschnittlich ≈ 40 ms gemessen werden (Anhang, Listing 12). Hier addieren sich sämtliche Latenzen in der Audioverarbeitung, deren Hauptanteil durch die gewählte Blockgröße des `convolve~` - Objektes bestimmt wird. Da es beim Überschreiten des Geofences für den Nutzer oder die Nutzerin keine anderen Rückkopplungen als den Start der Audioperformance gibt, sind nur die durch das Headtracking auftretenden Latenzen und die der sich anschließend angesteuerten Objekte perzeptiv relevant. Eine potentiell wahrnehmbare Latenz setzt sich somit aus der benötigten Zeit für das Sammeln und Verarbeiten der Headtracking-Daten und der von `earplug~` für die HRTF-Faltung beanspruchten Zeit zusammen. Diese kann mit $0,5$ ms vernachlässigt werden, während die Latenz der Bestimmung der Azimut- und Elevationdaten mit $20 - 40$ ms deutlich unter den oben genannten Grenzwerten bleibt.

[1] Vgl. Tabelle 4

6 Fazit

Ziel dieser Arbeit war die Konzeption und Implementierung einer mobilen iOS-App für geolokalisierte dynamische Binauralsynthese. Dazu wurden zu Beginn die akustischen und psychoakustischen Merkmale des räumlichen Hörens im teilbegrenzten Halbfreifeld betrachtet, um daraus Anforderungen an die Virtualisierung abzuleiten. Nach umfassender Recherche der Hardware-Ressourcen des iPhones 5s erfolgte die Implementierung als native App in Objective-C. Die DBS wurde in der grafischen echtzeitorientierten Audio-Programmiersprache Pure Data implementiert[1].

Abb. 29: App-Icon

Die Demonstrations-App *Binaural Soundmap* liegt als Beta-Version vor. Mit ihr ist möglich, virtuelle Klangquellen, hier Soundspots genannt, in einer 2D-Kartenansicht zu platzieren und deren akustische Eigenschaften zu modifizieren. Die App auralisiert jeweils eine ruhende virtuelle Schallquelle entfernungs- und blickrichtungsabhängig. Die Performance startet sobald sich der Nutzer oder die Nutzerin innerhalb des Geofences des jeweiligen Soundspots befindet. Nach dem Verlassen der Region schaltet die Software einem Nearest-Neighbour-Kriterium folgend automatisch zum nächstgelegenen Soundspot um. Mehrere Soundspots sind kontextgebunden in der Oberkategorie der Soundwalks organisiert. Für die persistente Datenspeicherung der Soundwalk- und Soundspot-Objekte wurde eine SQLite-Datenbank mit Core Data angelegt.[2] Über das GUI sind die Soundwalk- und Soundspot-Objekte modifizierbar. Reverse Geocoding ergänzt automatisch angelegte Objekte mit Adressdaten. Über das GUI sind Geofence, Audiofile und mit der OIR auch der akustische Raum wählbar. UX-Elemente sind animierte Soundspot-Annotationen mit transparenten Overlays in der Kartendarstellung und eine animierte Bewegungsparallaxe zwischen Hintergrund und Bedienelementen.
Die Hörsituation wird unterteilt in Nah- und Fernfeld und deren Übergang ineinander linear interpoliert. Im Fernfeld werden frühe und späte Reflexionen durch eine Faltung mit prototypischen OIR abgebildet, während im Nahfeld das Direktsignal mit der zeitverzögerten Bodenreflexion als einzige frühe Reflexion überlagert wird. Diese unterliegt einer frequenz-, winkel- und impedanzabhängigen Dämpfung[3]. Dem Direktsignal und der Bodenreflexion werden mittels separater HRTF-Faltung richtungsspezifische Merkmale aufgeprägt. Der Pegel des Gesamtsignals wird mit -6 dB pro Entfernungsverdopplung reduziert. Ein High-Shelf-Filter approximiert die Dissipationsdämpfung der Luft[4] bei mittlerer Temperatur und Luftfeuchtigkeit. Der verfälschende Einfluss der unbekannten HpTF der Kopfhörer des Nutzers oder der Nutzerin wird durch eine partielle Entzerrung um das 3 kHz Peak reduziert.[5] Als Enhancement wurde ein blickwinkelabhängiger Notch-Filter hinzufügt, der seine maximale Filterwirkung bei $0°$ Azimut erreicht und dessen Ziel unter Ausnutzung des richtungsspezifischen Frequenzbandes bei $1,2$ kHz eine Reduzierung der FBA ist.[6] Als Konsequenz aus der Temperaturabhängigkeit der Schallgeschwindigkeit und der impedanzabhängigen Dämpfung der Bodenreflexion[7] wurden jahreszeitlich differenzierte OIRs [159] angeboten.

[1] Vgl. Abschnitt 4.6
[2] Vgl. Abschnitt 4.4 und Abbildung 17
[3] Vgl. Abschnitt 2.3.2 und 4.6.1
[4] Vgl. Abschnitt 2.3.1 Abbildung 3
[5] Vgl. Abschnitt 3.2.3 und 4.6.4
[6] Vgl. Abschnitt 4.6.5
[7] Vgl. Abschnitt 2.3.1

CPU- und Speicherauslastung bewegen sich trotz mehrfacher Verwendung rekursiver Filter und zwei simultanen direkten Faltungen in einem Rahmen, innerhalb dessen eine flüssige und unterbrechungsfreie Audiosynthese durchführbar ist. Unstetigkeiten seitens der GPS-basierten Ortung beeinträchtigen mit abnehmender Distanz zum Soundspot-Zentrum zunehmend die Konsistenz der wahrgenommene Schallereignisrichtung- und -entfernung.
Die von Sandvad [20, S. 9] und Mackensen [95, S. 45] ermittelten Grenzwerte für Headtracking- und Verarbeitungslatenzen 96 ms bzw. 85 ms werden in dieser Beta-Version mit ≈ 40 ms deutlich unterschritten. Damit hat sich gezeigt, dass eine geolokalisierte dynamische Binauralsynthese mit den genannten Einschränkungen und Features auf dem iPhone 5s und unter Verwendung von Pure Data möglich ist.

7 Ausblick

Optimierungen hinsichtlich des auffälligen Problems der unsteten und relativ groben Ortung sind für eine Konsistenz der wahrgenommenen Schallereignisrichtung und -entfernung notwendige nächste Schritte. Eine Möglichkeit wäre die Verwendung des in Abschnitt 4.3 erwähnten GDGPS mit einer um mehr als das Zehnfache verbesserten Ortungsgenauigkeit von bis zu ± 0.3 m. Da diese Dienste bislang nur kostenpflichtig angeboten werden und die communitybasierte Lösung mit OpenDGPS [161] noch nicht verfügbar ist, könnte die Verwendung von GPS auf die Region außerhalb eines Geofences beschränkt werden. Diese könnte in einem von der horizontalen Genauigkeit bestimmten Abstand vom Geofence stattfinden, da dort auch bei Ortungsunstetigkeiten keine Audiosynthese stattfindet. Innerhalb des Soundspots könnte die Positionsbestimmung durch eine passive, inertiale Navigation ersetzt werden. Solche kombinierte Systeme werden bereits erfolgreich getestet [96], müssen aber aufgrund der sich mit der Zeit kumulierenden Sensordrift in regelmäßigen Abständen kalibriert werden. Diese Kalibrierung könnte aufgrund der geringen Ausdehnung der Soundspots wie auch die GPS-Nutzung ausschließlich außerhalb des Geofences stattfinden. Desweiteren kann die notwendige Häufigkeit der Kalibrierung durch die Verwendung von Kalman-Filtern [97] sowie durch den Verzicht auf die driftanfälligen Gyroskope reduziert werden [98].

Als Erweiterung ist eine skalierbare Abbildung mehrerer, auch bewegter Quellen denkbar. Durch die Verwendung von Unique-Identifiern ist es unter libpd möglich, von einen PD-Patch mehrere Instanzen aufzurufen. Dadurch wäre es möglich simultane Quellen abzubilden. Um dies auf der Smartphone-Hardware umzusetzen, muss jedoch der Audio-Algorithmus optimiert werden. Dies betrifft in erster Linie die HRTF-Faltung, deren Kosten ohne Einbuße in der Echtzeitfähigkeit reduziert werden müssten. Eine Optimierung der Segmentierten Schnellen Faltung mit variabler Blockgröße findet sich bei Garcia [99]. Er stellt fest, dass anstelle einer stetigen Segmentverbreiterung eine Kombination aus uniformer und variabler Blockgröße den Rechenaufwand deutlich senkt. Dabei kann während der Segmentierten Faltung mit uniformer Blockgröße das Spektrum dieser Blöcke als *Frequency–Domain Delay Line (FDL)* zusammenfassend gespeichert werden, um die anschließende IFFT für eine FDL als Ganzes, anstatt für jedes der wachsenden Segmente separat durchzuführen. Durch Kombination mehrerer FDLs erreicht Garcia eine Verbesserung der Effizienz um mehr als das Zweifache im Vergleich zum Gardner-Algorithmus. Interessant daran im Hinblick auf Binaural-Rendering ist, dass diese Optimierung auch die Anfangslast betrifft. [ebda.]

Weitere Performanceverbesserungen können potentiell durch variable Längen der BFs erreicht werden, die sich nach Anzahl der simulierten Quellen und Leistungsfähigkeit der Hardware [62, S. 128] richten. Da die Audioübertragung zwischen den modularen Pure-Data-Objekten im Zeitbereich erfolgt, ist mit Blick auf eine Echtzeit-Verarbeitung eine Transformation in den Frequenzbereich nur mit begrenzter Häufigkeit sinnvoll. Innerhalb eines jeden Objektes wird eine separate (I)FFT notwendig, wodurch sich deren blockgrößenbedingte Latenzen im PD-Patch addieren. Bei einer Umsetzung der DBS mit Pure Data könnte ein an die mobilen Bedürfnisse [100] angepasstes, skalierbares PD-External, das die CPU-kritischen Verarbeitungsprozesse der DBS nach Möglichkeit zusammengefasst in den Frequenzbereich verlagert und optimierte Faltungsalgorithmen verwendet, eine Performanceverbesserung und eine Steigerung der Zahl simultaner Quellen ermöglichen. Die zunehmende Verbreitung von libpd in mobilen Apps in Verbindung mit dem ho-

hen Interesse an mobilen AR-Inhalten würde die dabei eingegangenen Kompromisse hinsichtlich der Modularität rechtfertigen. Eine andere Möglichkeit wäre die Plattformunabhängigkeit der pd-basierten Audiosynthese zugunsten von mehr Freiheit in der Algorithmuserstellung aufzugeben. In diesem Fall könnte unter iOS das native Accelerate-Frameworks genutzt werden, das mit `vDSP` die in dieser App verwendeteten mathematischen Funktionen zur Vektor- und Matritzenarithmetik, Fouriertransformation, Faltung, biquadratischer Filterung und darüber hinaus Korrelation anbietet. [162] Denn durch eine mit der Entfernung zunehmenden Dekorrelation [101] der Ohrsignale kann die akustische Simulation der Distanz weiter verbessert werden. Gleichzeitig könnte in entgegengesetzter Richtung für den beim natürlichen Hören nicht vorkommenden Fall des physischen Durchschreitens einer Klangquelle durch zunehmende Korrelation zum Soundspot-Zentrum hin bewusst eine IHL für diesen Punkt herbeigeführt werden.
Nicht zuletzt könnte dem Nutzer oder der Nutzerin eine Auswahl verschiedener nicht-individueller HRTFs angeboten und das nachträgliche Hinzufügen eines eigenen individuellen HRTF-Sets ermöglicht werden. Unterstützt durch einen implementierten Hör- und Lokalisationstest könnte das binaurale Filterset ausgewählt werden, das für den Nutzer oder die Nutzerin die maximale Lokalisierungsleistung erbringt. Auf diese Art könnte eine überzeugende Performance auch für Menschen erzielt werden, deren Physiognomie von den nach Durchschnittsmaßen gefertigten DHs abweicht. Darüber hinaus könnten Ansätze zum HRTF-Enhancement aufgegriffen werden, wie die nachträgliche Individualisierung nicht-individueller HRTFs. Einer dieser Ansätze ist die individuelle Anpassung der ITDs nach Lindau et al. [102] durch die Verwendung eines minimalphasigen BRIR-Sets, bei dem die originalen ITDs entfernt wurden und mit einer in Abhängigkeit von den Kopfmaßen des Hörers parametrisierten variablen Delay-Line generiert werden. Lineare Zusammenhänge zwischen Pinnagröße und -form und spezifischen, für die Lokalisation der Elevation bedeutsamen Pinna-Notches untersuchten Hu et al. [104] mit statistischen Methoden, um darauf basierend eine Individualisierung nicht-individueller HRTFs zu erreichen. Selbst die Abbildung individueller persönlicher HRTFs ist u. U. verbesserbar. Gupta et al. [103] gingen Hinweisen für eine geringere FBA bei Hörern mit besonders hervorstehenden Außenohren [93] nach und ermittelten spektrale Profile, die übertragen auf eine persönliche HRTFs zu einer deutlichen Verminderung der FBA und einer allgemeinen Steigerung der Lokalisationsleistung der Versuchspersonen führte. Eine komplette Synthese individueller HRTFs auf Grundlage biometrischer Daten ist Ziel von Forschungen von Sottek und Genuit [105].
Als Anwendungsbereiche von Weiterentwicklungen der Demonstrations-App *Binaural Soundmap* könnten explorative Stadtführungen, interaktive (Hör-)Spiele oder Infotainment-Anwendungen fokussiert werden. Eine zusätzliche Schnittstelle für die Anbindung eines externen Headtrackers oder die Einbindung einer vorhandenen API eines sensorbestückten Headsets [163] wäre ein nächster Schritt in diese Richtung.

Literatur

Normen

[1] ISO (2001): „DIN 18005-1: 2000-12 (Entwurf): Schallschutz."

[2] ISO (1996): „ISO 9613-1, Acoustics - Attenuation of sound during propagation outdoors - Part 1: Calculation of the absorption of sound by the atmosphere."

[3] ISO (1996): „ISO 9613-2 : Dämpfung des Schalls bei der Ausbreitung im Freien."

[4] ITU (1990): „ITU-R BS.708, Determination of the electro-acoustical properties of studio monitor headphones."

Publikationen

[5] Weiser, Mark (1995): „Human-computer Interaction." Kap. The Computer for the 21st Century. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., S. 933–940.

[6] Sutherland, Ivan E. (1965): „The Ultimate Display." In: *Proceedings of IFIP Congress*. S. 506–508.

[7] Dörner, Ralf; Wolfgang Broll; Paul Grimm und Bernhard Jung (2014): *Virtual und augmented reality (VR/AR): Grundlagen und Methoden der Virtuellen und Augmentierten Realität*. Springer-Verlag.

[8] Azuma, Ronald (1997): „A Survey of Augmented Reality." In: *Presence: Teleoperators and Virtual Environments*, vol. 6. S. 355–385.

[9] Vorländer, Michael (2008): *Auralization*. Springer Berlin Heidelberg.

[10] Zwicker, E. (1982): *Psychoakustik*. Berlin Heidelberg New York: Springer Verlag.

[11] Macpherson, Ewan A. und John C. Middlebrooks (2002): „Listener weighting of cues for lateral angle: The duplex theory of sound localization revisited." In: *The Journal of the Acoustical Society of America*, 111(5) S. 2219.

[12] Slavik, Karl M. und Stefan Weinzierl (2008): „Handbuch der Audiotechnik." In: Stefan Weinzierl (Hrg.) *Wiedergabeverfahren*, Kap. 11. Springer-Verlag, S. 610–647.

[13] Madole, Dave und Durand Begault (1995): „3-D Sound for Virtual Reality and Multimedia." In: *Computer Music Journal*, 19(4) S. 99.

[14] Blauert, Jens und Jonas Braasch (2008): „Räumliches Hören." In: Stefan Weinzierl (Hrg.) *Handbuch der Audiotechnik*, Kap. 3. Springer-Verlag, S. 87–121.

[15] Seeber, Bernhard (2002): *Untersuchung der auditiven Lokalisation mit einer Lichtzeigermethode*. Ph.D. thesis, Technische Universität München.

[16] Lindau, Alexander et al. (2014): „A Spatial Audio Quality Inventory (SAQI).” In: *Acta Acustica united with Acustica*, 100(5) S. 984–994.

[17] Haas, Helmut (1951): „Über den Einfluss eines Einfach-Echos auf die Hörsamkeit von Sprache.” In: *Acustica 1* S. 49–58.

[18] (2006): *Akustische Wellen und Felder*, 101. Berlin: Deutsche Gesellschaft für Akustik e.V.

[19] Inanaga, Kiyofumi; Yuji Yamada und Hiroshi Koizumi (1995): „Headphone System with Out-of-Head Localization Applying Dynamic HRTF (Head-Related Transfer Function).” In: *Audio Engineering Society Convention 98*.

[20] Sandvad, J. (1996): „Dynamic Aspects of Auditory Virtual Environments.” In: *Audio Engineering Society Convention 100*.

[21] Minnaar, Pauli; Søren Krarup Olesen andFlemming Christensen und Henrik Møller (2001): „The importance of head movements for binaural room synthesis.” International Conference on Auditory Display, Finnland: Department of Acoustics, Aalborg University.

[22] Lindau, Alexander (2006): *Ein Instrument zur softwaregestützten Messung binauraler Raumimpulsantworten in mehreren Freiheitsgraden*. Masterarbeit, Technische Universität Berlin.

[23] Xiang, Pei; Miller Puckette und David Camargo (2005): „Experiments On Spatial Gestures In Binaural Sound Display.” In: *Proceedings of ICAD*. International Conference on Auditory Display.

[24] Zahorik, Pavel (2002): „Auditory display of sound source distance.” In: *In: Proceedings of the 2002 International Conference on Auditory Displays*. S. 326–332.

[25] Plenge, G. (1972): „Über das Problem der Im-Kopf-Lokalisation.” In: *Mitteilungen aus dem Heinrich-Hertz-Institut*, vol. 26 von *5*. HHI Berlin-Charlottenburg, S. 274–252.

[26] Trahiotis, Constantine; Leslie R. Bernstein; Richard M. Stern und Thomas N. Buell (2005): „Interaural Correlation as the Basis of a Working Model of Binaural Processing: An Introduction.” In: *Sound Source Localization*, 25. Springer-Verlag, S. 238–271.

[27] Jeffress, Lloyd A. (1948): „A place theory of sound localization.” In: *Journal of Comparative and Physiological Psychology*, 41(1) S. 35–39.

[28] Dau, Torsten (2002): „Modelle der auditorischen Verarbeitung.” In: N Dillier (Hrg.) *5. Jahrestagung der Deutschen Gesellschaft für Audiologie*. Zürich: DGA, S. CD–ROM.

[29] Rossing, Thomas D. et al. (2007): *Springer Handbook of Acoustics*. Springer.

[30] Harris, C.M.; Columbia University und George C. Marshall Space Flight Center (1967): *Absorption of Sound in Air Versus Humidity and Temperature*. Nr. Bd. 647 in Absorption of Sound in Air Versus Humidity and Temperature. National Aeronautics and Space Administration.

[31] Görne, T. (2014): *Tontechnik: Hören, Schallwandler, Impulsantwort und Faltung, digitale Signale, Mehrkanaltechnik, tontechnische Praxis*. Carl Hanser Verlag GmbH & Company KG.

[32] Brunner, Stefan (2006): *Perzeptive Bewertung von kammfilterartigen linearen Verzerrungen in der Audiotechnik*. Masterarbeit, Technische Universität Berlin.

[33] Müller, G. und M. Möser (2013): *Taschenbuch der Technischen Akustik*. Springer Berlin Heidelberg.

[34] Theile, Günther (1980): *Über die Lokalisation im überlagerten Schallfeld*. Ph.D. thesis, Technische Universität Berlin.

[35] Barth, M. (2009): *Akustische Tomographie zur zeitgleichen Erfassung von Temperatur- und Strömungsfeldern*. Wissenschaftliche Mitteilungen aus dem Institut für Meteorologie der Universität Leipzig und dem Institut für Troposphärenforschung e. V. Leipzig. Institut für Meteorologie der Universität Leipzig.

[36] Farnell, Andrew James (2012): *Designing sound : procedural audio research based on the book by Andy Farnell*. Ph.D. thesis, Escola das Artes, Universidade Católica Portuguesa. URL `http://hdl.handle.net/10400.14/13552`.

[37] Zhou, X. und X. Zha (2004): *Bestimmung des Absorptionsgrades bei schrägem Schalleinfall*. IBP-Mitteilung: Fraunhofer-Institut für Bauphysik. Fraunhofer-Institut für Bauphysik.

[38] Sinambari, G.R. und S. Sentpali (2014): *Ingenieurakustik: Physikalische Grundlagen und Anwendungsbeispiele*. Springer Fachmedien Wiesbaden.

[39] Piercy, J. E. (1977): „Review of noise propagation in the atmosphere." In: *The Journal of the Acoustical Society of America*, 61(6) S. 1403. doi:10.1121/1.381455. URL `http://dx.doi.org/10.1121/1.381455`.

[40] Goossens, Sebastian (2010): *Simulation und Auralisierung kleiner Räume*. Tech. rep., Institut für Rundfunktechnik, München.

[41] Kruse, Roland; Ping Rong und Volker Mellert (2007): „Messung der Bodenimpedanz im Freifeld." In: *Fortschritte der Akustik*. Stuttgart, S. 499 – 501.

[42] Delany, M.E. und E.N. Bazley (1970): „Acoustical properties of fibrous absorbent materials." In: *Applied Acoustics*, 3(2) S. 105–116.

[43] Embleton, T. F. W. (1983): „Effective flow resistivity of ground surfaces determined by acoustical measurements." In: *The Journal of the Acoustical Society of America*, 74(4) S. 1239.

[44] Attenborough, K.; K.M. Li und K. Horoshenkov (2006): *Predicting Outdoor Sound*. Taylor & Francis.

[45] Stumpner, Roman und Sebastian Goossens (2004): „About the influence of the reflection coefficient on simulation in room acoustics." In: *Fortschritte der Akustik*. Institut für Rundfunktechnik, Strasbourg: Deutschen Arbeitsgemeinschaft für Akustik.

[46] Miki, Yasushi (1990): „Acoustical properties of porous materials. Modifications of Delany-Bazley models." In: *Journal of the Acoustical Society of Japan (E)*, 11(1) S. 19–24.

[47] León, Fernando Puente; Uwe Kiencke und Holger Jäkel (2011): *Signale und Systeme*. 5. München: Oldenbourg Verlag.

[48] Zölzer, Udo (2008): „Digitale Signalverarbeitung, Filter und Effekte." In: Stefan Weinzierl (Hrg.) *Handbuch der Audiotechnik*, Kap. 15. Springer-Verlag, S. 813–847.

[49] Cooley, James W.; Peter A.W. Lewis und Peter D. Welch (1967): „Application of the fast Fourier transform to computation of Fourier integrals, Fourier series, and convolution integrals." In: *Audio and Electroacoustics, IEEE Transactions on*, 15(2) S. 79–84. doi: 10.1109/TAU.1967.1161904.

[50] Kammeyer, Karl-Dirk und Kristian Korschel (2006): *Digitale Signalverarbeitung: Filterung und Spektralanalyse*, vol. 6. Wiesbaden: Vieweg + Teubner.

[51] Weinstein, C. J. (1969): *Quantization Effects in Digital Filters*. Tech. Rep. 468, Lincoln Laboratory MIT, Lexington, Massachusetts.

[52] Gauss, C. F. (1866): „Theoria interpolationis methodo nova tractata."

[53] Vorländer, M. (2003): *Akustische Virtuelle Realität*. Institut für Technische Akustik.

[54] Weinzierl, Stefan (2008): „Handbuch der Audiotechnik." In: Stefan Weinzierl (Hrg.) *Grundlagen*, Kap. 1. Springer-Verlag, S. 1–40.

[55] Orfanidis, S.J. (1996): *Introduction to Signal Processing*. Prentice-Hall.

[56] Brigham, Elbert O. (1997): *FFT-Anwendungen*. Oldenbourg Verlag.

[57] Gardner, William G. (1995): „Effizient Convolution without Input-Output Delay." In: *Journal Audio Enginering*, vol. 43. AES, S. 127–136.

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

[59] Nicol, R. (2010): *Binaural Technology*. Monographs of the American Ethnological Society. Audio Engineering Society.

[60] Fellner, Maria und Robert Höldrich (1998): „Außenohr-Übertragungsfunktion - Messung und Datensätze." In: *IEM-Report*.

[61] Blauert, J. (1974): *Räumliches Hören*. Nr. Bd. 1 in Monographien der Nachrichtentechnik. Hirzel.

[62] Begault, Durand (2000): „3-D Sound for Virtual Reality and Multimedia."

[63] Estrella, Jorgos (2011): *Real Time Individualization Of Interaural Time Differences For Dynamic Binaural Synthesis*. Masterarbeit, Technische Universität Berlin.

[64] Lokki, Tapio und Ville Pulkki (2002): „Evaluation of Geometry-based Parametric Auralization." In: *Audio Engineering Society Conference: 22nd International Conference: Virtual, Synthetic, and Entertainment Audio*.

[65] Krokstad, A.; S. Strom und S. Sørsdal (1968): „Calculating the acoustical room response by the use of a ray tracing technique." In: *Journal of Sound and Vibration*, 8(1) S. 118 – 125.

[66] Gamper, Hannes und Tapio Lokki (2011): „Instant BRIR Acquisition for Auditory Events in Audio Augmented Reality Using Finger Snaps." In: *Principles And Applications Of Spatial Hearing*. World Scientific Pub Co Pte Lt, S. 1–4.

[67] Svensson, Peter (2008): „Auralization. Fundmamentals of Acoustics, Modelling, Simulation, Algorithms and Acoustic Virtual Reality by Michael Vorländer." In: *Acta Acustica united with Acustica*, 94(4) S. 643–643.

[68] Schärer, Zora und Alexander Lindau (2008): „Evaluation von Verfahren zur Entzerrung von binauralen Signalen." In: *TONMEISTERTAGUNG*, vol. 25. Leipzig: Verband Deutscher Tonmeister e.V., S. 515–532.

[69] Schaerer, Zora (2008): *Kompensation von Frequenzgängen im Kontext der Binauraltechnik*. Masterarbeit, Technische Universität Berlin.

[70] 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. URL `http://www.aes.org/e-lib/browse.cfm?elib=16166`.

[71] Theile, Günther (1986): „On the Standardization of the Frequency Response of High-Quality Studio Headphones." In: *J. Audio Eng. Soc*, 34(12) S. 956–969.

[72] Dickreiter, M.; V. Dittel; W. Hoeg und M. Wöhr (2014): *Handbuch der Tonstudiotechnik*. De Gruyter.

[73] Møller, Henrik; Clemen Boje Jensen; Dorte Hammershøi und Michael Friis Sørensen (1995): „Design Criteria for Headphones." In: *J. Audio Eng. Soc*, 43(4) S. 218–232.

[74] Lorho, Gaëtan (2009): „Subjective Evaluation of Headphone Target Frequency Responses." In: *Audio Engineering Society Convention 126*.

[75] Hess, Wolfgang (2012): „Head-Tracking Techniques for Virtual Acoustics Applications." In: *Audio Engineering Society Convention 133*.

[76] Furht, Borko (Hrg.) (2011): *Handbook of Augmented Reality*. Springer.

[77] Milgrim, Paul; Haruo Takemura; Akira Utsumi und Fumio Kishino (1995): „Augmented reality: a class of displays on the reality-virtuality continuum." In: Hari Das (Hrg.) *Telemanipulator and Telepresence Technologies*, vol. 2351. Boston,MA: The International Society for Optical Engineering.

[78] Tönges, Raffael (2015): *An Augmented Acoustics Demonstrator With Realtime Stereo Up-Mixing And Binaural Auralization*. Masters thesis, Technische Universität Berlin.

[79] Brinkmann, Peter et al. (2011): „Embedding Pure Data with libpd." In: *in Proceedings of the Pure Data Convention, Weimar*.

[80] Brinkmann, Peter (2012): *Making Musical Apps*. O'REILLY.

[81] Puckette, Miller (1996): „Pure Data: another integrated computer music environment." In: *in Proceedings, International Computer Music Conference*. S. 37–41.

[82] Puckette, Miller (2007): *The Theory and Technique of Electronic Music*. World Scientific Press.

[83] James, Dick (2013): *Inside Today's Systems & Chips: A Survey of the Past Year*. Tech. rep., Chipworks Confidential.

[84] Lundberg, Kent H. (2002): „Analog-to-Digital Converter Testing." Unpublished Material.

[85] Apple Inc. (2015): *iPhone 5s Benutzerhandbuch*. Cupertino, CA.

[86] Ozyagcilar, Talat (2013): *Calibrating an eCompass in the Presence of Hard and Soft-Iron Interference*. Application Note AN4246, Freescale Semiconductor.

[87] Mineta, T et al. (1996): „Three-axis capacitive accelerometer with uniform axial sensitivities." In: *J. Micromech. Microeng.*, 6(4) S. 431–435.

[88] Bosch (2011): „BMA220 - Digital, triaxial acceleration sensor." Data sheet.

[89] STMicroelectronics (2011): *Everything about STMicroelectronics' 3-axis digital MEMS*. Tech. rep., STMicroelectronics.

[90] Grohn, Matti; Tapio Lokki und Tapio Takala (2002): „Static and Dynamic Sound Source Localization in a Virtual Room." In: *Audio Engineering Society Conference: 22nd International Conference: Virtual, Synthetic, and Entertainment Audio*.

[91] Gardner, William G. (1995): „HRTF measurements of a KEMAR." In: *The Journal of the Acoustical Society of America*, 97(6) S. 3907.

[92] Bücklein, Roland (1981): „The Audibility of Frequency Response Irregularities." In: *J. Audio Eng. Soc*, 29(3) S. 126–131.

[93] Burkhard, M. D. (1975): „Anthropometric manikin for acoustic research." In: *The Journal of the Acoustical Society of America*, 58(1) S. 214.

[94] Maxwell, R. J. und M. D. Burkhard (1979): „Larger ear replica for KEMAR manikin." In: *The Journal of the Acoustical Society of America*, 65(4) S. 1055.

[95] Mackensen, Philip (2004): „Auditive Localization. Head movements, an additional cue in Localization."

[96] Soloviev, Andrey; Charles Toth und Dorota Grejner-Brzezinska (2012): „Performance of deeply integrated GPS/INS in dense forestry areas." In: *Journal of Applied Geodesy*, 6(1).

[97] Gathmann, Felix; Christian Dernehl; Dominik Franke und Stefan Kowalewski (2012): „An integrated vision aided GPS/INS Navigation System for ultra-low-cost MAVs." In: *International Micro Air Vehicle Conference (IMAV)*. IMAV, S. 1 – 8.

[98] Marques Filho, Edmundo A; Helio Koiti Kuga und Atair Rios Neto (2006): „Integrated GPS/INS Navigation System Based on a Gyroscope-Free IMU." In: *Brazilian Conference on Dynam-ics, Control and Their Applications*, vol. 5. S. 1322–1327.

[99] Garcia, Guillermo (2002): „Optimal Filter Partition for Efficient Convolution with Short Input/Output Delay." In: *Audio Engineering Society Convention 113*.

[100] Sander, Christian; Frank Wefers und Dieter Leckschat (2012): „Scalable Binaural Synthesis on Mobile Devices." In: *Audio Engineering Society Convention 133*.

[101] Borß, C. und R. Martin (2009): „An Improved Parametric Model for Perception-Based Design of Virtual Acoustics." In: *AES 35th Int. Conference*. London, UK.

[102] Lindau, Alexander; Jorgos Estrella und Stefan Weinzierl (2010): „Individualization of Dynamic Binaural Synthesis by Real Time Manipulation of ITD." In: *Audio Engineering Society Convention 128*.

[103] Gupta; Barreto und Ordonez (2002): „Spectral modification of head-related transfer functions for improved virtual sound spatialization." In: *IEEE International Conference on Acoustics Speech and Signal Processing*. Institute of Electrical & Electronics Engineers (IEEE).

[104] Hu, Hongmei; Liang Chen und Zhen yang Wu (2008): „The Estimation of Personalized HRTFs in Individual VAS." In: *2008 Fourth International Conference on Natural Computation*. Institute of Electrical & Electronics Engineers (IEEE).

[105] Sottek, Roland und Klaus Genuit (1999): „Physical modeling of individual head-related transfer functions (HRTFs)." In: *The Journal of the Acoustical Society of America*, 105(2) S. 1162.

Internetquellen

[106] TNS Infratest, BVDW, Google (2014): „Global Connected Consumer Studie." URL http://services.google.com/fh/files/misc/global-connected-consumer-studie-deutschland.pdf.

[107] Respondi AG (2013): „BVMI / PLAYFAIR: Studie zur mobilen Musiknutzung." URL http://www.playfair.org/fileadmin/user_upload/downloads/BVMI_PLAYFAIR_Studie_zur_mobilen_Musiknutzung_131206.pdf.

[108] Moravec, Hans (1998): „ROBOT: Mere Machine to Transcendent Mind." URL https://www.frc.ri.cmu.edu/~hpm/book97/ch3/processor.list.txt.

[109] Lindau, Alexander (2014): „A Spatial Audio Quality Inventory for Virtual Acoustic Environments (SAQI)." URL https://speakerdeck.com/alindau/a-spatial-audio-quality-inventory-for-virtual-acousticenvironments-saqi.

[110] Lamancusa, J. S. (2000): „NOISE CONTROL - Outdoor Sound Propagation." URL http://www.me.psu.edu/lamancusa/me458/10_osp.pdf.

[111] Weisstein, Eric W. (2015): „Danielson-Lanczos Lemma." URL http://mathworld.wolfram.com/Danielson-LanczosLemma.html.

[112] Fauske, Kjell Magne (2006): „Radix-2 FFT signal flow." URL http://www.texample.net/tikz/examples/radix2fft/.

[113] Smith, Steven (2002): „Digital Signal Processing: A Practical Guide for Engineers and Scientists." URL http://www.dspguide.com/pdfbook.htm.

[114] Gardner, Bill und Keith Martin (1994): „HRTF Measurements of a KEMAR Dummy-Head Microphone." URL http://sound.media.mit.edu/resources/KEMAR.html.

[115] Synthtopia (2011): „It Is Not Possible To Play Music On An Android Phone." URL http://www.synthtopia.com/content/2011/12/03/developer-it-is-not-possible-to-play-music-on-an-android-phone/.

[116] IK Multimedia (2015): „Announcing iGrand Piano and iLectric Piano for Android devices." URL http://www.ikmultimedia.com/news/?item_id=6929.

[117] IEM - Institut für Elektronische Musik und Akustik (2003): „Pure Data." URL https://puredata.info.

[118] Remote Control Productions (2012): „Inception." URL https://itunes.apple.com/de/app/inception-the-app/id405235483?mt=8.

[119] Gameloft (2015): „The Dark Knights Rises." URL https://itunes.apple.com/de/app/the-dark-knight-rises/id522704697?mt=8.

[120] Lucky Frame Limited (2013): „Bad Hotel." URL https://itunes.apple.com/de/app/bad-hotel/id548665066?mt=8.

[121] Sandler, Seth (2015): „NodeBeat." URL https://itunes.apple.com/de/app/nodebeat-playful-music-for/id428440804?mt=8.

[122] Lucky Frame Limited (2012): „Pugs Luv Beats." URL https://itunes.apple.com/lu/app/pugs-luv-beats/id488112663?l=de&mt=8.

[123] Mischke, Alfred (2014): „Vermessungskunde, Teil 7: GNSS." URL http://www.ruhr-uni-bochum.de/geodaesie/download/Skript%20Teil%207%20-%20GNSS.pdf.

[124] U.S. Air Force (2014): „GPS.GOV." URL http://www.gps.gov/systems/gps/performance/accuracy/.

[125] Tanner, Jason; Jim Morrison; Dick James; Ray Fontaine und Phil Gamache (2013): „Inside the iPhone 5s." URL http://www.chipworks.com/about-chipworks/overview/blog/inside-the-iphone-5s/.

[126] AKM (2013): „AK8963, 3-axis Electronic Compass." URL http://www.akm.com/akm/en/file/datasheet/AK8963.pdf.

[127] Konvalin, Christopher (2009): „Motion/Velocity/Displacement – Compensating for Tilt, Hard-Iron, and Soft-Iron Effects." URL http://www.sensorsmag.com/sensors/motion-velocity-displacement/compensating-tilt-hard-iron-and-soft-iron-effects-6475.

[128] VectorNav Technologies (2012): „Magnetometer." URL http://www.vectornav.com/support/library/magnetometer.

[129] Intantech (2013): „Application Note: Accelerometer Calibration." URL http://www.intantech.com/files/Intan_RHD2000_accelerometer_calibration.pdf.

[130] Research und Markets (2013): „iPhone 5S MEMS Gyroscope STMicroelectronics 3x3mm - Reverse Costing Analysis." URL http://www.researchandmarkets.com/reports/2673993/.

[131] STMicroelectronics (2013): „L3GD20H – MEMS motion sensor: three-axis digital output gyroscope." URL http://www.st.com/st-web-ui/static/active/en/resource/technical/document/datasheet/DM00060659.pdf.

[132] Apple Inc. (2015): „iPhone and iPad: Calibrating Compass." URL https://support.apple.com/en-us/HT203301.

[133] Apple Inc. (2013): „Core Location Framework Reference." URL https://developer.apple.com/library/ios/documentation/CoreLocation/Reference/CoreLocation_Framework/.

[134] Senatsverwaltung für Stadtentwicklung und Umwelt Abteilung III - Geoinformation Senatsverwaltung für Stadtentwicklung und Umwelt, Abteilung III - Geoinformation, Referat III B - Geodätische Referenzsysteme (2010): „Vorschriftensammlung für das Vermessungswesen." URL http://www.stadtentwicklung.berlin.de/geoinformation/sapos/.

[135] Senatsverwaltung für Stadtentwicklung und Umwelt (2010): „Wie genau arbeitet mein GNSS/GPSEmpfänger?" URL http://www.stadtentwicklung.berlin.de/geoinformation/bezugssysteme/download/flyer_wie_genau_arbeitet_mein_gps.pdf.

[136] Apple Inc. (2013): „Core Motion Framework Reference." URL https://developer.apple.com/library/ios/documentation/CoreMotion/Reference/CoreMotion_Reference/.

[137] Yoaz Bar-Sever, Jet Propulsion Laboratory California Institute of Technology (2013): „The Global Diffential GPS System." URL http://www.gdgps.net.

[138] Apple Inc. (2013): „Cocoa Core Competencies: Model-View-Controller." URL https://developer.apple.com/library/ios/documentation/General/Conceptual/DevPedia-CocoaCore/MVC.html.

[139] Reenskaug, Trygve (1979): „Models - Views - Controllers." URL http://folk.uio.no/trygver/2007/MVC_Originals.pdf.

[140] Apple Inc. (2015): „Core Data Programming Guide." URL https://developer.apple.com/library/mac/documentation/Cocoa/Conceptual/CoreData/cdProgrammingGuide.html.

[141] Apple Inc. (2015): „MKMapView Class Reference." URL https://developer.apple.com/library/prerelease/ios/documentation/MapKit/Reference/MKMapView_Class/index.html#//apple_ref/occ/cl/MKMapView.

[142] Apple Inc. (2015): „NSTableView Class Reference." URL https://developer.apple.com/library/prerelease/mac/documentation/Cocoa/Reference/ApplicationKit/Classes/NSTableView_Class/index.html#//apple_ref/occ/cl/NSTableView.

[143] Apple Inc. (2013): URL https://developer.apple.com/library/prerelease/ios/documentation/UIKit/Reference/UIPickerView_Class/index.html#//apple_ref/occ/cl/UIPickerView.

[144] Apple Inc. (2013): „UITextView Class Reference." URL https://developer.apple.com/library/prerelease/ios/documentation/UIKit/Reference/UITextView_Class/.

[145] Apple Inc. (2011): „UISlider Class Reference." URL https://developer.apple.com/library/prerelease/ios/documentation/UIKit/Reference/UISlider_Class/.

[146] Apple Inc. (2013): „UIButton Class Reference." URL https://developer.apple.com/library/prerelease/ios/documentation/UIKit/Reference/UIButton_Class/.

[147] Apple Inc. (2015): „UILabel Class Reference." URL https://developer.apple.com/library/prerelease/ios/documentation/UIKit/Reference/UILabel_Class/.

[148] Apple Inc. (2014): „NSManagedObjectContext Class Reference." URL https://developer.apple.com/library/prerelease/watchos/documentation/Cocoa/Reference/CoreDataFramework/Classes/NSManagedObjectContext_Class/.

[149] Apple Inc. (2013): „NSFetchedResultsController Class Reference." URL https://developer.apple.com/library/prerelease/watchos/documentation/CoreData/Reference/NSFetchedResultsController_Class/.

[150] Apple Inc. (2013): „UITableViewCell Class Reference." URL https://developer.apple.com/library/prerelease/ios/documentation/UIKit/Reference/UITableViewCell_Class/.

[151] Apple Inc. (2015): „CMPedometer Class Reference." URL https://developer.apple.com/library/prerelease/watchos/documentation/CoreMotion/Reference/CMPedometer_class/.

[152] Apple Inc. (2014): „NSOperationQueue Class Reference." URL https://developer.apple.com/library/mac/documentation/Cocoa/Reference/NSOperationQueue_class/.

[153] Moser-Booth, Mike (2011): „biquad.mmb .pd." URL https://github.com/dotmmb/mmb/blob/master/biquad.mmb%7E.pd.

[154] Geiger, Günter (2000): „highshelf." URL https://svn.code.sf.net/p/pure-data/svn/trunk/externals/ggee/filters/highshelf.c.

[155] Bristow-Johnson, Robert (1998): „Cookbook formulae for audio EQ biquad filter coefficients." URL http://www.musicdsp.org/files/Audio-EQ-Cookbook.txt.

[156] Statistisches Bundesamt (2013): „Körpermaße nach Altersgruppen: Männer." URL https://www.destatis.de/DE/ZahlenFakten/GesellschaftStaat/Gesundheit/GesundheitszustandRelevantesVerhalten/Tabellen/KoerpermasseInsgesamt.html.

[157] Statistisches Bundesamt (2013): „Körpermaße nach Altersgruppen: Frauen." URL https://www.destatis.de/DE/ZahlenFakten/GesellschaftStaat/Gesundheit/GesundheitszustandRelevantesVerhalten/Tabellen/KoerpermasseFrauen.html.

[158] Brent, William (2010): „WilliamBrent.com." URL http://williambrent.conflations.com/pages/research.html.

[159] Boom Library (2013): „OUTDOOR IMPULSE RESPONSES." URL http://www.boomlibrary.com/boomlibrary/products/outdoor-impulse-responses.

[160] Geiger, Günter (2000): „equalizer." URL https://svn.code.sf.net/p/pure-data/svn/trunk/externals/ggee/filters/equalizer.c.

[161] OpenDGPS (2012): „OpenDGPS - Differential GPS for the rest of us." URL http://opendgps.de/about/.

[162] Apple Inc. (Aug): „Accelerate Framework Reference – vDSP Reference." URL https://developer.apple.com/library/prerelease/ios/documentation/Accelerate/Reference/vDSPRef/index.html#//apple_ref/doc/uid/TP40009464.

[163] GN Store Nord (2013): „Intelligent Headset." URL https://intelligentheadset.com/developer/.

Anhang

Ortungsgenauigkeit A-GPS

Listing 9: Ortungsgenauigkeit des iPhone 5s am geodätischen Referenzpunkt Berlin Mitte

```
[...]
-[SoundMapVC getDistanceFromSoundSpot]  currentLongitude: 13.371683  ,
    currentLatitude: 52.522586
-[SoundMapVC locationManager:didUpdateLocations:] didUpdateLocation
-[SoundMapVC observeValueForKeyPath:ofObject:change:context:]
    horizontalAccuracy: +/-5m
-[SoundMapVC getDirectionToCurrentSoundSpot]  degree: 139.433060
-[SoundMapVC observeValueForKeyPath:ofObject:change:context:] new speed: 0mps
-[SoundMapVC getDistanceFromSoundSpot]  currentLongitude: 13.371683  ,
    currentLatitude: 52.522586
-[SoundMapVC locationManager:didUpdateLocations:] didUpdateLocation
-[SoundMapVC observeValueForKeyPath:ofObject:change:context:]
    horizontalAccuracy: +/-5m
-[SoundMapVC getDirectionToCurrentSoundSpot]  degree: 139.433060
-[SoundMapVC observeValueForKeyPath:ofObject:change:context:] new speed: 0mps
-[SoundMapVC getDistanceFromSoundSpot]  currentLongitude: 13.371683  ,
    currentLatitude: 52.522586
-[SoundMapVC locationManager:didUpdateLocations:] didUpdateLocation
-[SoundMapVC locationManager:didUpdateHeading:] cleanHeading: 64.0 degree
-[SoundMapVC locationManager:didUpdateHeading:] heading: mag: 200.5 degree /
    true: 204.0 degree
-[SoundMapVC observeValueForKeyPath:ofObject:change:context:]
    horizontalAccuracy: +/-5m
-[SoundMapVC getDirectionToCurrentSoundSpot]  degree: 139.433060
-[SoundMapVC observeValueForKeyPath:ofObject:change:context:] new speed: 0mps
-[SoundMapVC getDistanceFromSoundSpot]  currentLongitude: 13.371683  ,
    currentLatitude: 52.522586
-[SoundMapVC locationManager:didUpdateLocations:] didUpdateLocation
-[SoundMapVC observeValueForKeyPath:ofObject:change:context:]
    horizontalAccuracy: +/-5m
-[SoundMapVC getDirectionToCurrentSoundSpot]  degree: 139.433060
-[SoundMapVC observeValueForKeyPath:ofObject:change:context:] new speed: 0mps
-[SoundMapVC getDistanceFromSoundSpot]  currentLongitude: 13.371683  ,
    currentLatitude: 52.522586
-[SoundMapVC locationManager:didUpdateLocations:] didUpdateLocation
-[SoundMapVC locationManager:didUpdateHeading:] cleanHeading: 65.0 degree
-[SoundMapVC locationManager:didUpdateHeading:] heading: mag: 201.5 degree /
    true: 205.0 degree
-[SoundMapVC observeValueForKeyPath:ofObject:change:context:]
    horizontalAccuracy: +/-5m
-[SoundMapVC getDirectionToCurrentSoundSpot]  degree: 139.433060
-[SoundMapVC observeValueForKeyPath:ofObject:change:context:] new speed: 0mps
-[SoundMapVC getDistanceFromSoundSpot]  currentLongitude: 13.371683  ,
    currentLatitude: 52.522586
-[SoundMapVC locationManager:didUpdateLocations:] didUpdateLocation
```

```
-[SoundMapVC observeValueForKeyPath:ofObject:change:context:] horizontalAccuracy: +/-5m
-[SoundMapVC getDirectionToCurrentSoundSpot]  degree: 139.433060
-[SoundMapVC observeValueForKeyPath:ofObject:change:context:] new speed: 0mps
-[SoundMapVC getDistanceFromSoundSpot]  currentLongitude: 13.371683  , currentLatitude: 52.522586
-[SoundMapVC locationManager:didUpdateLocations:] didUpdateLocation
-[SoundMapVC observeValueForKeyPath:ofObject:change:context:] horizontalAccuracy: +/-5m
-[SoundMapVC getDirectionToCurrentSoundSpot]  degree: 139.433060
-[SoundMapVC observeValueForKeyPath:ofObject:change:context:] new speed: 0mps
-[SoundMapVC getDistanceFromSoundSpot]  currentLongitude: 13.371683  , currentLatitude: 52.522586
-[SoundMapVC locationManager:didUpdateLocations:] didUpdateLocation
-[SoundMapVC observeValueForKeyPath:ofObject:change:context:] horizontalAccuracy: +/-5m
-[SoundMapVC getDirectionToCurrentSoundSpot]  degree: 139.433060
-[SoundMapVC observeValueForKeyPath:ofObject:change:context:] new speed: 0mps
-[SoundMapVC getDistanceFromSoundSpot]  currentLongitude: 13.371683  , currentLatitude: 52.522586
-[SoundMapVC locationManager:didUpdateLocations:] didUpdateLocation
-[SoundMapVC observeValueForKeyPath:ofObject:change:context:] horizontalAccuracy: +/-5m
-[SoundMapVC getDirectionToCurrentSoundSpot]  degree: 139.433060
-[SoundMapVC observeValueForKeyPath:ofObject:change:context:] new speed: 0mps
-[SoundMapVC getDistanceFromSoundSpot]  currentLongitude: 13.371683  , currentLatitude: 52.522586
[...]
```

Latenzen

Listing 10: Latenz Azimuthberechnung

```
2015-08-31 20:49:50.492 BinauralSoundMap[417:48066] azimuth: 305 deg
2015-08-31 20:49:50.633 BinauralSoundMap[417:48066] azimuth: 304 deg
2015-08-31 20:49:50.731 BinauralSoundMap[417:48066] azimuth: 305 deg
2015-08-31 20:49:50.773 BinauralSoundMap[417:48066] azimuth: 303 deg
2015-08-31 20:49:50.793 BinauralSoundMap[417:48066] azimuth: 302 deg
2015-08-31 20:49:50.813 BinauralSoundMap[417:48066] azimuth: 298 deg
2015-08-31 20:49:50.834 BinauralSoundMap[417:48066] azimuth: 295 deg
2015-08-31 20:49:50.854 BinauralSoundMap[417:48066] azimuth: 293 deg
2015-08-31 20:49:50.874 BinauralSoundMap[417:48066] azimuth: 290 deg
2015-08-31 20:49:50.893 BinauralSoundMap[417:48066] azimuth: 289 deg
```

(Test auf dem iPhone 5s mit *Binaural Soundmap*)

Listing 11: Latenz der Elevationsberechnung

```
2015-09-01 00:32:38.459 BinauralSoundMap[493:78601] elevation: 42 deg
2015-09-01 00:32:38.493 BinauralSoundMap[493:78601] elevation: 40 deg
2015-09-01 00:32:38.536 BinauralSoundMap[493:78601] elevation: 39 deg
```

```
2015-09-01 00:32:38.570 BinauralSoundMap[493:78601] elevation: 37 deg
2015-09-01 00:32:38.614 BinauralSoundMap[493:78601] elevation: 35 deg
2015-09-01 00:32:38.647 BinauralSoundMap[493:78601] elevation: 34 deg
2015-09-01 00:32:38.690 BinauralSoundMap[493:78601] elevation: 32 deg
2015-09-01 00:32:38.724 BinauralSoundMap[493:78601] elevation: 30 deg
2015-09-01 00:32:38.768 BinauralSoundMap[493:78601] elevation: 29 deg
2015-09-01 00:32:38.801 BinauralSoundMap[493:78601] elevation: 27 deg
```

(Test auf dem iPhone 5s mit *Binaural Soundmap*)

Listing 12: Systemlatenz der App

```
2015-09-01 00:44:28.302 BinauralSoundMap[521:82264] Pd: TICK: bang
2015-09-01 00:44:28.303 BinauralSoundMap[521:82264] azimuth: 222 deg
2015-09-01 00:44:28.331 BinauralSoundMap[521:82264] azimuth: 221 deg
2015-09-01 00:44:28.360 BinauralSoundMap[521:82264] Pd: TACK: 29.6012
[...]
2015-09-01 00:49:04.012 BinauralSoundMap[534:83722] Pd: TICK: bang
2015-09-01 00:49:04.012 BinauralSoundMap[534:83722] azimuth: 203 deg
2015-09-01 00:49:04.060 BinauralSoundMap[534:83722] Pd: TACK: 24.8652
[...]
2015-09-01 00:50:00.841 BinauralSoundMap[534:83722] Pd: TICK: bang
2015-09-01 00:50:00.982 BinauralSoundMap[534:83722] Pd: TACK: 17.3097
[...]
2015-09-01 00:50:54.194 BinauralSoundMap[534:83722] Pd: TICK: bang
2015-09-01 00:50:54.194 BinauralSoundMap[534:83722] azimuth: 204 deg
2015-09-01 00:50:54.207 BinauralSoundMap[534:83722] azimuth: 203 deg
2015-09-01 00:50:54.239 BinauralSoundMap[534:83722] Pd: TACK: 4.08251
[...]
2015-09-01 00:51:58.734 BinauralSoundMap[534:83722] Pd: TICK: bang
2015-09-01 00:51:58.747 BinauralSoundMap[534:83722] Pd: TACK: 18.9136
```

(TICK = Start, TACK = erstes Audiosample am Output, Test auf dem iPhone 5s mit *Binaural Soundmap*)

Listing 13: Latenz des PD-Objektes `earplug~`

```
earplug~ took time: 0.000501s
earplug~ took time: 0.000501s
earplug~ took time: 0.000501s
earplug~ took time: 0.000501s
earplug~ took time: 0.000501s
earplug~ took time: 0.000501s
earplug~ took time: 0.000501s
earplug~ took time: 0.000501s
earplug~ took time: 0.000501s
```

(Test auf dem iPhone 5s mit *Binaural Soundmap*)

Listing 14: Latenz des PD-Objektes `convolve~`

```
convolve~ took time: 0.002512s
convolve~ took time: 0.003231s
convolve~ took time: 0.000502s
```

```
convolve~ took time: 0.002510s
convolve~ took time: 0.000064s
convolve~ took time: 0.000503s
convolve~ took time: 0.003068s
convolve~ took time: 0.003291s
convolve~ took time: 0.001003s
convolve~ took time: 0.001002s
```

(Test auf dem iPhone 5s mit *Binaural Soundmap*)