



Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Bachelorarbeit

Marvin Geitner

**Erweiterung der Wellenfeldsynthese-Software WONDER um
3D-Quellenpositionen und zusätzliche
Lautsprecherkomponenten**

*Fakultät Technik und Informatik
Studiendepartment Informatik*

*Faculty of Engineering and Computer Science
Department of Computer Science*

Marvin Geitner

**Erweiterung der Wellenfeldsynthese-Software WONDER um
3D-Quellenpositionen und zusätzliche
Lautsprecherkomponenten**

Bachelorarbeit eingereicht im Rahmen der Bachelorprüfung

im Studiengang Bachelor of Science Technische Informatik
am Department Informatik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer: Prof. Dr. Wolfgang Fohl
Zweitgutachter: Prof. Dr.-Ing. Andreas Meisel

Eingereicht am: 15. Mai 2015

Marvin Geitner

Thema der Arbeit

Erweiterung der Wellenfeldsynthese-Software WONDER um 3D-Quellenpositionen und zusätzliche Lautsprecherkomponenten

Stichworte

Wellenfeldsynthese, Quellenposition, Schallquelle, Open Sound Control, 3D-Audio, WONDER

Kurzzusammenfassung

In dieser Arbeit wird die Wellenfeldsynthese-Software WONDER derart erweitert, dass eine 3D-Quellenpositionierung möglich ist. Des Weiteren werden vier zusätzliche Lautsprecherkomponenten in das bestehende System der Wellenfeldsynthese-Anlage eingebunden, um einen dreidimensionalen räumlichen Höreindruck zu schaffen. Basierend auf einem in dieser Arbeit entwickelten Verfahren, werden die entstehenden Schallfelder simuliert und die Resultate diskutiert. Um den Höreindruck einer dreidimensional positionierten Schallquelle zu ermitteln, wird ein Hörtest durchgeführt und ausgewertet.

Marvin Geitner

Title of the paper

Extension of the wave field synthesis software WONDER to 3D sources positions and additional loudspeaker components

Keywords

Wave Field Synthesis, Sourceposition, Soundsource, Open Sound Control, 3D-Audio, WONDER

Abstract

This thesis describes a extension of the wave field synthesis software WONDER with the aim of a 3D source positioning. Further four additional loudspeaker components are added into the existing wave field synthesis system to create a three dimensional spatial sound. Based on a method developed in this work, the resulting sound fields are simulated and the results are discussed. There will be a hearingtest to determine the auditory impression of a three dimensional positioned soundsource.

Inhaltsverzeichnis

1. Einleitung	1
1.1. Ziel der Arbeit	1
1.2. Themenabgrenzung	1
1.3. Aufbau	2
2. Wellenfeldsynthese	3
2.1. Stereofonie	3
2.2. Das Prinzip der Wellenfeldsynthese	3
2.3. Quellentypen	5
2.3.1. Punktquelle	5
2.3.2. Linearquelle	5
2.3.3. Fokussierte Quelle	5
2.4. Einschränkung der Wellenfeldsynthese	6
2.5. Das Wellenfeldsynthese-Labor der HAW	7
2.5.1. Die Wellenfeldsynthese-Anlage	7
2.5.2. Erweiterung der Wellenfeldsynthese-Anlage	11
3. Die Wellenfeldsynthese-Software WONDER	12
3.1. Aufbau von WONDER	12
3.2. Konfiguration von WONDER	13
3.3. SuperCollider	14
3.4. Erweiterung von WONDER	14
3.4.1. Vorbereitung	14
3.4.1.1. Cwonder	15
3.4.1.2. Twonder	16
3.4.1.3. Zweidimensionale Quellenpositionierung	17
3.4.1.4. Änderungen der Software	18
3.4.2. Implementierung der Änderungen	19
3.4.2.1. Hinzufügen zusätzlicher OSC-Nachrichten	19
3.4.2.2. Hinzufügen zusätzlicher Callback-Funktionen	20
3.4.2.3. Anpassen der Datentypen	21
3.4.2.4. Anpassen der Konfiguration	22
3.4.2.5. Kompatibilität	24
3.4.2.6. Einschränkung der Nutzung	25
3.4.3. Ergebnis	25

4. Simulation des Schallfeldes	26
4.1. Aufbau	26
4.2. Resultate	30
4.2.1. Simulation der oberen Lautsprecherebene	30
4.2.2. Hinzufügen der Zusatzlautsprecher	30
4.2.3. Höhenveränderung der Schallquelle	33
4.2.4. Positionsabhängigkeit des Hörers	35
4.3. Diskussion der Resultate	38
5. Ergebnisse	43
5.1. Vergleich der Simulation mit den Messdaten	43
5.1.1. Durchführung der Messung	43
5.1.2. Auswertung der Messdaten	44
5.2. Hörtest für 3D-Quellenpositionen	47
5.2.1. Versuchsdurchführung	47
5.2.2. Auswertung der Ergebnisse	51
6. Fazit	54
6.1. Zusammenfassung	54
6.2. Fazit	54
6.3. Ausblick	55
A. Benutzertest und Messpunkte	56
A.1. SuperCollider Skript	58
A.2. Messpunkte	61
A.3. Daten der Probanden	63
B. Matlab-Code	68
B.1. Simulation	68
B.2. Auswertung Messdaten	73
Literaturverzeichnis	78

1. Einleitung

Das optimale Hörereignis ist bei herkömmlicher Wiedergabe von Tonaufnahmen für den Zuhörer auf einen bestimmten Bereich beschränkt. Wird dieser Bereich verlassen, geht die räumliche Darstellung des akustischen Ereignisses verloren. Die Wellenfeldsynthese ist ein räumliches Audiowiedergabeverfahren, das es ermöglicht, innerhalb des Hörfeldes einen räumlichen Eindruck zu schaffen und so eine natürliche akustische Umgebung für den Hörer zu produzieren. Durch den Einsatz speziell für die Wellenfeldsynthese entwickelter Software wird ein Schallfeld synthetisiert, das identisch zum originalen Schallfeld ist. Somit wird ein Höreindruck geschaffen, der dem realen Klangereignis sehr nahe kommt.

1.1. Ziel der Arbeit

Ziel dieser Arbeit ist die Erweiterung der Wellenfeldsynthese-Software WONDER¹ um 3D-Quellenpositionen und zusätzliche Lautsprecherkomponenten. Hierbei wird die Software in der Version von Carola Christoffel [Chr14] entsprechend erweitert, dass eine dreidimensionale Positionierung einer Schallquelle möglich ist. Des Weiteren werden vier zusätzliche Lautsprecher in das System der Wellenfeldsynthese-Anlage eingebunden, um einen dreidimensionalen Höreindruck zu schaffen. Mit Hilfe von Matlab wird ein Verfahren entwickelt, das die entstehenden Schallfelder simuliert. Im Vorfeld dieser Arbeit war bereits klar, dass theoretisch betrachtet ein dreidimensionaler Höreindruck mit gegebenen Mitteln nicht erreicht werden kann. Anhand der praktischen Umsetzung soll dies in der vorliegenden Arbeit überprüft werden.

1.2. Themenabgrenzung

Diese Arbeit befasst sich mit der Erweiterung der Wellenfeldsynthese-Software WONDER um 3D-Quellenpositionen. Hierbei wird die Software in bestimmten Bereichen erweitert, jedoch bleiben das allgemeine Prinzip der Wellenfeldsynthese und der Algorithmus zur Berechnung

¹<http://www.sourceforge.net/projects/swonder>

der Audiosignale bestehen und werden nicht verändert. Die Implementierung einer grafischen Oberfläche zur Darstellung der 3D-Quellenpositionen wird in dieser Arbeit nicht berücksichtigt.

1.3. Aufbau

Kapitel 2 gibt einen Überblick über das Prinzip der Wellenfeldsynthese und stellt anhand einer Grafik die Synthese einer Wellenfront dar. Des Weiteren wird ein kurzer Überblick über die unterschiedlichen Arten einer Schallquelle und die Einschränkungen bei der Wellenfeldsynthese gegeben.

In Kapitel 3 wird die Wellenfeldsynthese-Software WONDER vorgestellt. Anschließend wird auf die für eine Erweiterung nötigen Komponenten der Software eingegangen, die Erweiterungen werden erfasst und implementiert.

Die Umsetzung des Verfahrens zur Simulation der Schallfelder erfolgt in Kapitel 4. Hier wird der mathematische Hintergrund des Verfahrens erläutert und die resultierenden Schallfelder werden beschrieben und diskutiert.

Kapitel 5 beschreibt die Messung eines erzeugten Schallfeldes im Vergleich zu einem simulierten Schallfeld sowie den Hörtest zur Wahrnehmung des tatsächlich entstehenden Höreindrucks. Anschließend wird das Ergebnis des Hörtests ausgewertet.

Zuletzt folgen eine Zusammenfassung sowie ein Fazit und ein kurzer Ausblick.

2. Wellenfeldsynthese

Die Wellenfeldsynthese ist ein räumliches Audiowiedergabeverfahren, mit dem es möglich ist, Wellenfronten virtueller Schallquellen zu synthetisieren und somit einen virtuellen akustischen Raum zu schaffen. Dieses Kapitel beschreibt das Prinzip der Wellenfeldsynthese und gibt einen Überblick über die Installation einer Wellenfeldsynthese-Anlage im Labor der HAW Hamburg.

2.1. Stereophonie

Die Stereophonie basiert auf dem psychoakustischen Prinzip der Phantomschallquelle¹. Bei der Stereophonie war es erstmals möglich, akustische Signale auf der Achse zwischen zwei Lautsprechern in Form einer Phantomschallquelle darzustellen. Durch unterschiedliche Pegeldifferenzen der Audiosignale, die durch zwei oder mehrere Lautsprecher wiedergegeben werden, ist es möglich, einen räumlichen Schalleindruck zu vermitteln. Je nach Pegelunterschied des Audiosignals der unterschiedlichen Lautsprecher wandert die entstehende Phantomschallquelle zu dem Lautsprecher, der den höheren Pegel aufweist. Haben beide Lautsprecher den selben Pegel, liegt das Signal direkt mittig zwischen den Lautsprechern[Ahr12].

Ein bedeutender Nachteil dieses Verfahrens ist, dass die räumliche Wahrnehmung von einem bestimmten Punkt abhängig ist, dem sogenannten Sweet Spot. Bewegt sich der Zuhörer aus diesem Bereich, ist eine korrekte räumliche Wahrnehmung der Schallquelle nicht mehr gegeben.

2.2. Das Prinzip der Wellenfeldsynthese

Bei der Wellenfeldsynthese beruht die Wahrnehmung nicht mehr auf dem psychoakustischen Prinzip der Phantomschallquelle, wie es bei der Stereophonie der Fall ist, sondern auf dem Huygens'schen Prinzip [The80]. Dieses Prinzip besagt, dass jeder Punkt einer Wellenfront als Ausgangspunkt einer neuen Welle, der sogenannten Elementarwelle, betrachtet werden kann. Die neue Lage der Wellenfront ergibt sich dabei durch Überlagerung sämtlicher Elementarwellen.

¹Dieses Prinzip tritt auf, wenn mehrere Schallereignisse einem gemeinsamen Hörereignis zugeordnet sind, sodass Hörereignisort und Schallquellenorte nicht übereinstimmen [The80].

Ziel der Wellenfeldsynthese ist es, ein reales Schallfeld möglichst exakt abzubilden. Um ein entsprechendes Schallfeld zu erzeugen, wird eine Anzahl von Lautsprechern in möglichst geringen Abständen um ein Hörfeld herum angeordnet. Das Schallfeld entwickelt sich dabei durch Laufzeitdifferenzen der Audiosignale für die einzelnen Lautsprecher. Nach Huygens ist es möglich, eine Wellenfront durch Überlagerung vieler Elementarwellen darzustellen und somit ein beliebiges Wellenfeld zu synthetisieren. Dieses Prinzip macht sich die Wellenfeldsynthese zunutze. Wird eine reale Schallquelle hinter den Lautsprechern platziert, breiten sich um diese Schallquelle kugelförmige Wellen gleichmäßig in alle Richtungen aus. Wird eine virtuelle Schallquelle an derselben Position platziert, berechnet das System für jeden Lautsprecher ein bestimmtes Signal und den Zeitpunkt, wann dieser aktiviert wird. Jeder Lautsprecher erzeugt somit eine Elementarwelle, deren Überlagerung eine Wellenfront bildet, die den Eindruck erweckt, als befände sich hinter den Lautsprechern eine identische reale Schallquelle [Chr14]. Da es sich hierbei um ein synthetisiertes Schallfeld handelt, spricht man von virtuellen Schallquellen. Abbildung 2.1 zeigt eine virtuelle Schallquelle, anhand der die Synthese einer Wellenfront demonstriert wird.

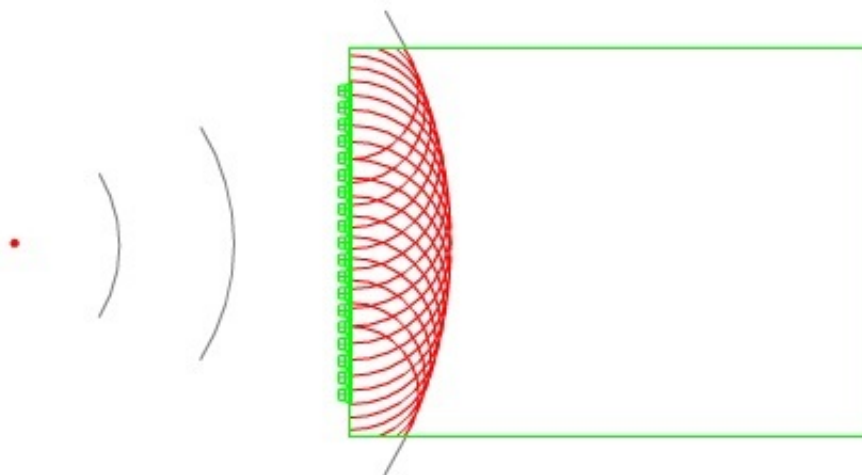


Abbildung 2.1.: Synthese einer Wellenfront [Aud]

Die als Punkt dargestellte virtuelle Schallquelle strahlt eine virtuelle Welle ab. Diese Welle trifft nach einer bestimmten Zeit auf die Rückseite der Lautsprecher. Für jeden dieser Lautsprecher wird nun eine entsprechende Elementarwelle berechnet, die sich im Raum fortsetzt. Durch die Überlagerung der Elementarwellen der einzelnen Lautsprecher bildet sich eine Wellenfront, die jener Welle entspricht, die von der virtuellen Schallquelle abgestrahlt wurde.

Da durch diesen Vorgang das komplette Schallfeld synthetisiert wird, ist es möglich, die virtuelle Schallquelle im gesamten Hörfeld exakt zu lokalisieren. Die Beschränkung der räumlichen Wahrnehmung auf einen Sweet Spot existiert somit bei der Wellenfeldsynthese nicht.

2.3. Quellentypen

Durch Wellenfeldsynthese lassen sich drei verschiedene virtuelle Quellentypen erzeugen: Punktquellen, Linearquellen und fokussierte Quellen.

2.3.1. Punktquelle

Virtuelle Schallquellen, die hinter den Lautsprechern positioniert sind, werden als Punktquellen bezeichnet. Punktquellen strahlen, wie in Abbildung 2.1 dargestellt, kugelförmige bzw. in der zweidimensionalen Darstellung kreisförmige Wellen ab. Punktquellen lassen sich innerhalb des Hörfeldes immer an einem bestimmten Ort lokalisieren.

2.3.2. Linearquelle

Die von einer Linearquelle abgestrahlten Wellen breiten sich nicht wie bei einer Punktquelle kugelförmig aus, sondern bilden ein ebenes Wellenfeld. Dieses Wellenfeld bildet sich, wenn sich eine virtuelle Schallquelle in einem sehr großen Abstand hinter den Lautsprechern befindet. Theoretisch betrachtet ist es möglich, eine Punktquelle in einem unendlich großen Abstand hinter den Lautsprechern zu positionieren. Die von dieser Quelle abgestrahlten Wellen gehen mit größer werdendem Abstand zur Quelle in ein immer ebener werdendes Wellenfeld über. Die Linearquelle lässt sich innerhalb des Hörfeldes mit gleicher Lautstärke und immer aus derselben Richtung wahrnehmen, ist jedoch nicht an einem bestimmten Ort lokalisierbar.

2.3.3. Fokussierte Quelle

Virtuelle Schallquellen, die innerhalb des Hörfeldes bzw. vor den Lautsprechern positioniert sind, werden als fokussierte Quellen bezeichnet. Durch Überlagerung der Wellen innerhalb des Hörfeldes bildet sich eine virtuelle Schallquelle. Fokussierte Quellen lassen sich wie Punktquellen an einem bestimmten Ort lokalisieren. Befindet sich der Zuhörer jedoch zwischen den aktiven Lautsprechern und der Quelle, ist keine korrekte Ortung mehr möglich. Eine genauere Betrachtung fokussierter Schallquellen bietet die Bachelorarbeit von Carola Christoffel [Chr14]. Abbildung 2.2 zeigt die verschiedenen Quellentypen.

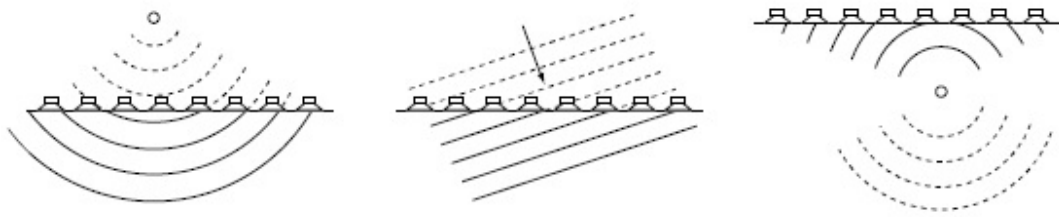


Abbildung 2.2.: Punktquelle, Linearquelle, Fokussierte Quelle [Mon11]

2.4. Einschränkung der Wellenfeldsynthese

In der Theorie der Wellenfeldsynthese wird von einem unendlichen kontinuierlichen Lautsprecherarray ausgegangen. Da diese Umsetzung in der Praxis jedoch nicht möglich ist, nähert man sich dem mit einer endlichen Anzahl an Lautsprechern mit gleichbleibenden Abständen untereinander an. Bedingt durch diese Abstände entsteht eine Grenzfrequenz, die sogenannte Aliasfrequenz. Wird die Aliasfrequenz von dem wiedergegebenen Audiosignal überschritten, treten Aliasing-Effekte auf, die zu räumlichen Abbildungsfehlern im reproduzierten Schallfeld führen [JD03].

Die meisten Systeme reduzieren die Wellenfeldsynthese auf die zweidimensionale Ebene. Da sich reale Schallwellen jedoch kugelförmig ausbreiten, würde sich das Verfahren theoretisch betrachtet auf eine dritte Dimension erweitern lassen. Gegen eine dritte Dimension spricht allerdings der hohe Rechen- und Hardwareaufwand. Bereits für eine zweidimensionale Wellenfeldsynthese wird bereits eine Vielzahl von Lautsprechern benötigt, für die für jede virtuelle Schallquelle die entsprechenden Signale berechnet werden müssen. Wird das Verfahren auf eine dritte Dimension erweitert, würden je nach Größe des Raumes entsprechend mehr Lautsprecher am System teilnehmen [Han14].

Eine Möglichkeit, virtuelle Schallquellen dennoch in der dritten Dimension darzustellen, beschreibt Matthew N. Montag in seiner Masterarbeit [Mon11]. Um die Einschränkung auf die horizontale und somit zweidimensionale Ebene aufzuheben, werden anstelle eines horizontal ausgerichteten Lautsprecherarrays zwei Lautsprecherarrays mit gleicher Anordnung auf verschiedenen Höhen positioniert. Wird eine virtuelle Schallquelle in horizontaler Richtung zwischen diesen Lautsprecherarrays positioniert, werden zunächst die Signale der Schallquelle auf der Höhe der Lautsprecherarrays berechnet. Abhängig vom horizontalen Abstand der Schallquelle zu den jeweiligen Lautsprecherarrays wird das Signal anschließend, ähnlich zur

Stereofonie, abgeschwächt bzw. verstärkt und somit eine räumliche Darstellung in dritter Dimension erreicht.

Eine Auflistung weiterer Einschränkungen findet sich in dem Paper *Wave Field Synthesis – A Promising Spatial Audio Rendering Concept* [The04].

2.5. Das Wellenfeldsynthese-Labor der HAW

An der HAW Hamburg besteht seit 2011 ein Wellenfeldsynthese-Labor, in dem eine von der Firma Four Audio² entwickelte Wellenfeldsynthese-Anlage installiert ist. Dieses Kapitel beschreibt das vorhandene System, den Aufbau und die für den Betrieb nötige Software. Außerdem wird die für diese Arbeit nötige Erweiterung der Wellenfeldsynthese-Anlage beschrieben.

2.5.1. Die Wellenfeldsynthese-Anlage

Die Wellenfeldsynthese-Anlage umfasst einen Bereich von etwa 5,0 x 5,7 m und besteht aus insgesamt 26 Lautsprechermodulen, wobei jeweils 6 Lautsprechermodule an den kurzen und jeweils 7 Lautsprechermodule an den langen Seiten angeordnet sind. Jedes Lautsprechermodul hat eine Gesamtlänge von 80 cm und besitzt jeweils 8 Wiedergabekanäle, was insgesamt eine Anzahl von 208 Wiedergabekanälen ergibt. Für den Betrieb der Wellenfeldsynthese-Anlage stehen 4 Rechner in einem 19“-Rack bereit. Dabei handelt es sich um einen OS-X-basierten Apple MacPro zur Bedienung der Wellenfeldsynthese-Anlage und drei Linux-basierte Rechner, die den Wellenfeldsynthese-Server und zwei Wellenfeldsynthese-Nodes darstellen und die Berechnung der Wellenfeldsynthese übernehmen [Foh13]. Zur Datenübertragung zwischen den Lautsprechermodulen und den Rechnern wird das von der Firma Audinate³ entwickelte Dante-Audionetzwerk verwendet. Das Dante-Audionetzwerk ermöglicht es, über Standardnetzwerkkomponenten sowohl die Audiodaten als auch beliebige andere Daten zu übertragen. Somit ist es möglich, die zur Steuerung nötigen Daten sowie die für die Lautsprechermodule berechneten Audiodaten über ein Netz zu übertragen. An diesem Netzwerk teilnehmende Rechner können Audiodaten entweder durch eine vorhandene Ethernet-Schnittstelle in Verbindung mit der virtuellen Soundkarte der Dante-Software oder über die Dante-PCIe-Soundkarte ausgeben [MGTK12].

Softwareseitig wird die frei verfügbare Software WONDER verwendet. WONDER besteht aus mehreren Komponenten, die sich auf mehrere Rechner verteilen lassen. In der Wellenfeldsynthese-Anlage der HAW kommen drei Komponenten der Software zum Einsatz. Einerseits c wonder,

²<http://www.fouraudio.com/>

³<https://www.audinate.com/>

das als Hauptbestandteil der Software auf dem Wellenfeldsynthese-Server läuft, andererseits twonder, das in mehreren Instanzen auf den beiden Wellenfeldsynthese-Nodes zur Berechnung der Audiosignale für die Lautsprechermodule dient. Als dritte Komponente stellt xwonder die grafische Oberfläche von WONDER dar und wird auf dem MacPro ausgeführt.

Zur Kommunikation unter den Komponenten verwendet WONDER Open Sound Control. Open Sound Control ist ein an der University of California entwickeltes Kommunikationsprotokoll. Es dient zur Kommunikation zwischen Computern, Sound-Synthesizern und anderen Multimedia-Systemen [WFM03]. Open Sound Control versendet Daten in Form von Nachrichten (OSC Messages), die nach einem bestimmten Aufbau gegliedert sind [Wri02]:

- Das **Address Pattern** ist ein OSC-String, das mit dem Zeichen „/“ beginnt, und gibt an, wohin die Nachricht gesendet werden soll. Die Open-Sound-Control-Adresse ist dabei hierarchisch aufgebaut, wie etwa bei einer URL oder einer Verzeichnisstruktur in einem Dateisystem.
- Der **Type Tag String** ist ein OSC-String, der mit dem Zeichen „“ beginnt, dem sich eine Zeichenfolge anschließt, die exakt die Reihenfolge der Argumententypen angibt. Jedes Zeichen dieses Strings stellt dabei den Typ des entsprechenden Arguments dar. Als Zeichen sind „i“, „f“, „s“ und „b“ verfügbar, welche die Datentypen int32, float32, String oder OSC-Blob beschreiben.
- Die **Arguments** enthalten die eigentlichen Daten, die übertragen werden sollen. Die Reihenfolge der Argumente muss dabei der Reihenfolge des Type Tag Strings entsprechen.

Abbildung 2.3 zeigt die implementierte Netzwerkstruktur der Wellenfeldsynthese-Anlage der HAW Hamburg.

Zum Starten oder Beenden der WONDER-Software werden verschiedene Skripte auf dem MacPro ausgeführt, die wiederum über den Remote-Zugriff Skripte auf dem Wellenfeldsynthese-Server und den Wellenfeldsynthese-Nodes ausführen. Außerdem befinden sich auf dem MacPro eine Reihe von Audio-Wiedergabeprogrammen, die dazu dienen, die Wellenfeldsynthese-Anlage mit Sound zu versorgen.

Des Weiteren ist ein Trackingsystem der Firma Advanced Realtime Tracking GmbH ⁴ Bestandteil der Wellenfeldsynthese-Anlage. Dieses System besteht aus mehreren Infrarotkameras, die auf den inneren Bereich der Wellenfeldsynthese-Anlage gerichtet sind.

⁴<http://www.ar-tracking.com/>

2. Wellenfeldsynthese

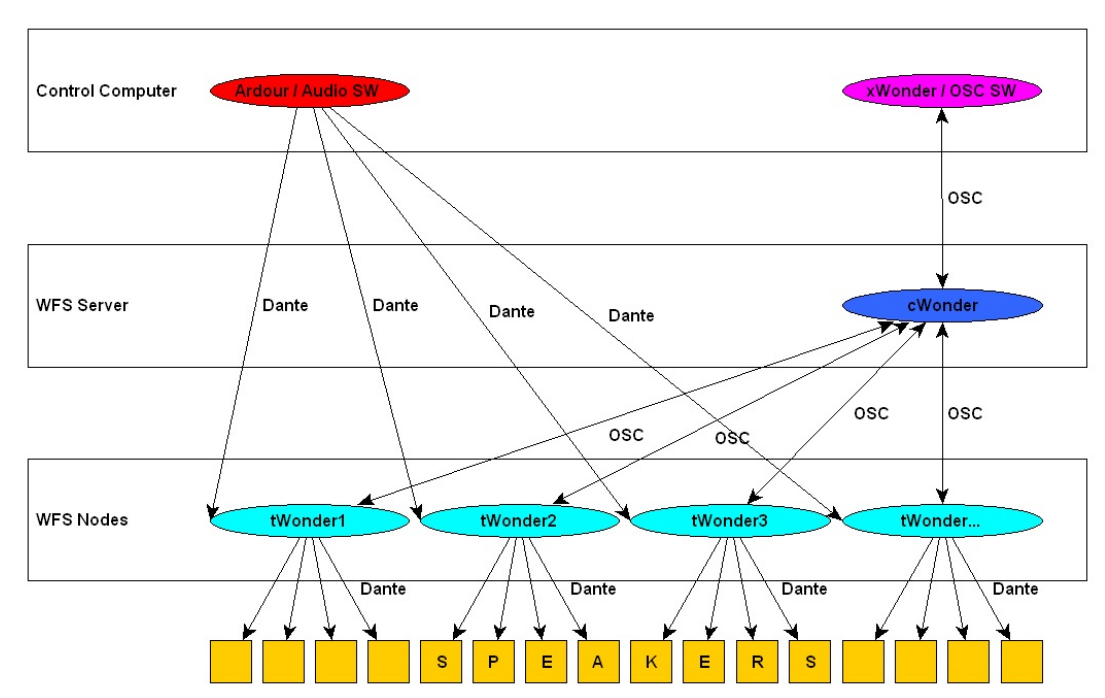


Abbildung 2.3.: Netzwerkstruktur der WFS-Anlage der HAW [MGTK12]

Bewegen sich die für dieses System entwickelten Marker innerhalb dieses Bereichs, lässt sich die Position einer oder mehrerer Marker anhand von Reflexionen erfassen. Mit diesem System lassen sich beispielsweise Klangquellen anhand einer gestengesteuerten Positionierung platzieren [Nog12]. Es ist außerdem eine Powerwall vorhanden, die sich aus sechs Displays zusammensetzt, die über einen separat gesteuerten Rechner als ein einziger Bildschirm fungieren. Die Powerwall wird in dieser Arbeit nicht genutzt. Abbildung 2.4 zeigt den Aufbau der Wellenfeldsynthese-Anlage mit den 26 Lautsprechermodulen, dem Trackingsystem und der Powerwall.

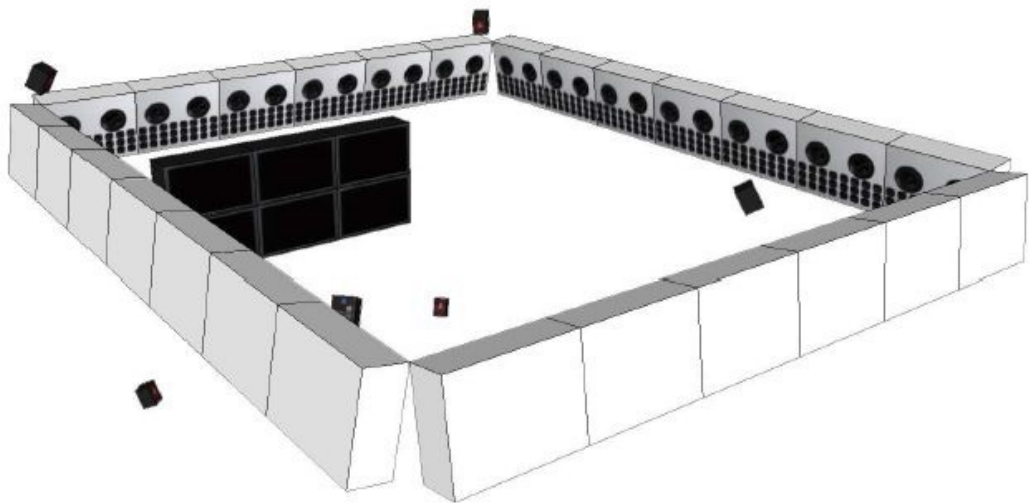


Abbildung 2.4.: Aufbau der WFS-Anlage der HAW [Nog12]

2.5.2. Erweiterung der Wellenfeldsynthese-Anlage

Für diese Arbeit ist es erforderlich, dass die bestehende Wellenfeldsynthese-Anlage um einige Hardware-Komponenten erweitert wird.

Es werden insgesamt vier 2-Wege-Aktiv-Bassreflexlautsprecher der Marke Adam A8X⁵ in das vorhandene System eingebunden. Zu erwähnen ist hierbei, dass die Adam-A8X-Lautsprecher ebenso wie die schon vorhandenen Lautsprechermodule über eine Netzwerkverbindung mit dem Dante-Audio-Netzwerk verbunden werden müssen. Standardmäßig besitzen die Adam-A8X-Lautsprecher jedoch keine netzwerkfähige Schnittstelle, sondern lediglich eine elektrische Steckverbindung des XLR-Standards. Um die Lautsprecher dennoch mit dem Audio-Netzwerk zu verbinden, wird ein RedNet-Ethernet-Netzwerk-Studio-Interface der Firma Focusrite⁶ genutzt. Dieses Interface bietet acht analoge Ein- und Ausgänge und ist somit in der Lage, acht Audiokanäle in das Audio-Netzwerk sowie aus dem Audio-Netzwerk zu übertragen. Das Interface wird dabei durch ein herkömmliches Netzkabel mit dem Dante-Audio-Switch verbunden und so dem Audio-Netzwerk zugänglich gemacht. Um nun die Adam-A8X-Lautsprecher in das Audio-Netzwerk zu integrieren, werden diese mittels einer DB25 auf einer XLR-Steckverbindung mit dem RedNet-Interface verbunden.

Damit die vier Lautsprecher über das Audio-Netzwerk angesteuert werden können, ist neben der hardwareseitigen auch eine softwareseitige Integration nötig. Hierzu stellt das Dante-Audio-Netzwerk einen Software-Controller bereit, mit dem es möglich ist, die Eingänge von sich im Netzwerk befindlichen Teilnehmern auf Ausgänge anderer Netzwerkteilnehmer abzubilden. Dieser Controller stellt auf einer grafischen Oberfläche alle sich im Audio-Netzwerk befindlichen Teilnehmer dar, wobei die Empfänger auf der vertikalen und die Sender auf der horizontalen Achse angeordnet sind.

Die vier Lautsprecher sind hier unter der Kennung des Interfaces zu finden und können somit im vorhandenen System genutzt werden.

⁵<http://www.adam-audio.com/>

⁶<http://www.focusrite.de/>

3. Die Wellenfeldsynthese-Software WONDER

WONDER ist eine Open-Source-Software zur Steuerung von Wellenfeldsynthese-Anlagen. Das Besondere dieser Software ist, dass es sich um eine verteilte Anwendung handelt. Dies ermöglicht es, die Komponenten der Software auf mehreren in einem Netzwerk befindlichen Rechnern laufen zu lassen und so die Rechenlast zu verteilen. In diesem Kapitel wird ein Überblick über WONDER gegeben und es werden die Erweiterungen der Software implementiert. Für eine detaillierte Beschreibung eignet sich die Dissertation von Marije Baalman [Baa07].

3.1. Aufbau von WONDER

WONDER lässt sich in drei elementare Komponenten gliedern:

- **Cwonder** Cwonder stellt den Mittelpunkt der Software dar und dient als Kontrolleinheit und Knotenpunkt. Die gesamte Kommunikation unter den Komponenten findet in cwonder statt und ist in unterschiedliche Streams gegliedert, sodass eine Komponente, die sich angemeldet hat, nur die für sich wichtigen Informationen erhält. Zur Kommunikation unter den Komponenten wird Open Sound Control verwendet. Zusätzliche Komponenten, die dem System beitreten, melden sich über Open-Sound-Control-Nachrichten bei cwonder an und erhalten bei etwaigen Änderungen eine entsprechende Benachrichtigung[BHKS07].
- **Twonder** Twonder stellt die Rendering-Einheit der Software dar. Twonder nutzt als Input Audiospuren externer Anwendungen und bildet diese auf virtuelle Schallquellen ab. Um die Wellenfront einer virtuellen Schallquelle zu synthetisieren, ist es notwendig, dass für jeden im System vorhandenen Lautsprecher die Verzögerung des Signals und die entsprechende Amplitude berechnet wird. Für die Berechnung der Verzögerung und der Amplitude nutzt twonder die Koordinaten der im System vorhandenen Lautsprecher und der virtuellen Schallquelle. Die Koordinaten der Lautsprecher lassen sich in einer entsprechenden Konfigurationsdatei eintragen.

- **Xwonder** Xwonder stellt die grafische Benutzeroberfläche der Software dar. Es bietet eine zweidimensionale Oberfläche, auf der der Benutzer in Echtzeit mit der Wellenfeldsynthese-Anlage interagieren kann [Baa07]. Das ermöglicht es, virtuelle Schallquellen zu erstellen und zu entfernen, deren Farbe zu ändern und Projekte anzulegen. Außerdem lässt sich die Position der Schallquellen durch einfaches Verschieben auf der Benutzeroberfläche verändern. Abbildung 3.1 zeigt die Benutzeroberfläche von xwonder.

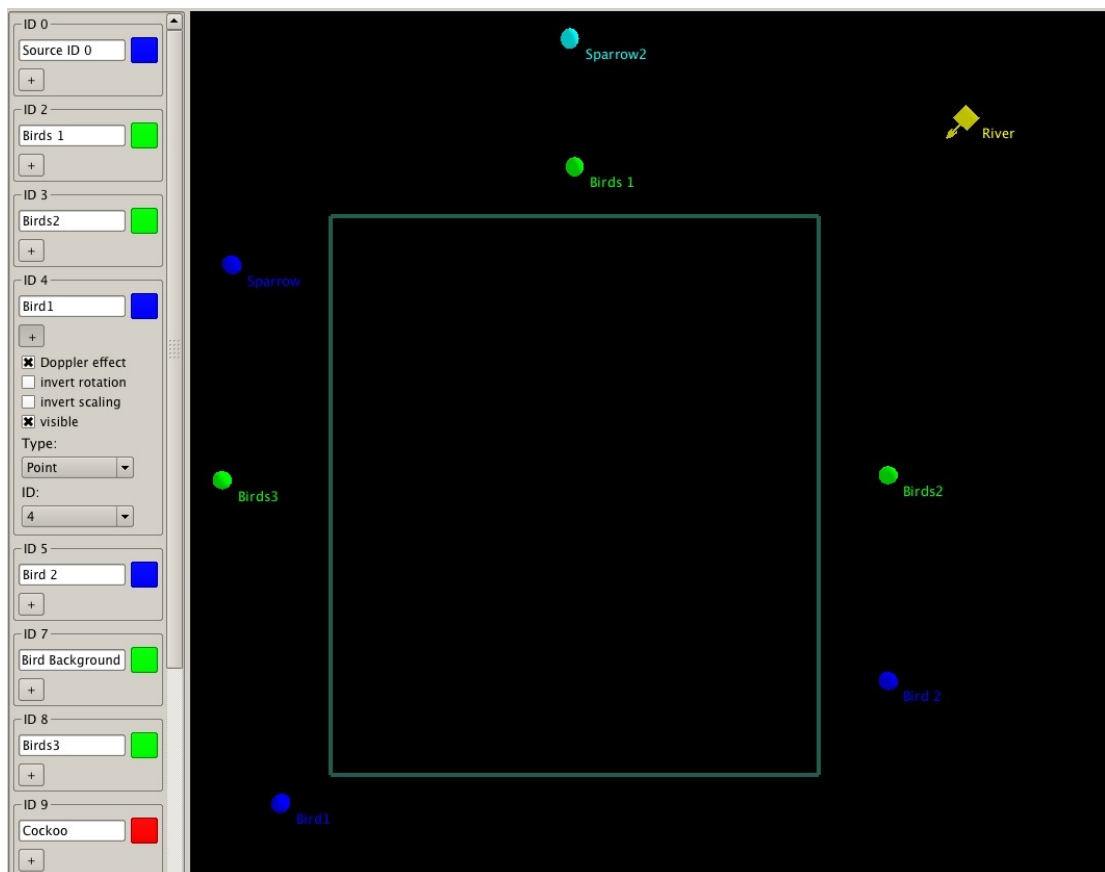


Abbildung 3.1.: Benutzeroberfläche von xwonder

3.2. Konfiguration von WONDER

WONDER lässt sich durch mehrere XML-Konfigurationsdateien an ein System anpassen. Die wichtigsten Konfigurationsdateien bestehen für die Komponenten cwonder und twonder. Beim Start der Software werden diese eingelesen und enthalten feste Werte. So enthält die

Konfigurationsdatei für cwonder die maximale Anzahl der Schallquellen sowie Name und Koordinaten des Raumes, der von der Wellenfeldsynthese-Anlage aufgespannt wird. Eine Besonderheit existiert für die Konfiguration von twonder. Die am System teilnehmenden Lautsprecher werden mit ihren Koordinaten, ihrer Spurenanzahl und Ausrichtung in separate XML-Dateien zusammengefasst und bilden somit Lautsprecher-Arrays. Diese Lautsprecher-Arrays werden genutzt, falls mehrere Instanzen von twonder gestartet werden. Jeder Instanz werden somit bestimmte Lautsprecher zur Verwaltung zugewiesen.

3.3. SuperCollider

SuperCollider¹ ist eine Programmierumgebung und Programmiersprache für Echtzeit-Audiosynthese und algorithmische Komposition und bietet die Möglichkeit, über OSC Messages die Software WONDER direkt anzusprechen. So ist es mit SuperCollider möglich, virtuelle Schallquellen zu erstellen, zu entfernen oder zu verschieben. Eine für WONDER spezifische Nachricht zur Verschiebung einer Schallquelle kann folgendermaßen aussehen:

```
NetAddr("192.168.56.10", 58100).  
  sendMsg("/WONDER/source/position", 1, 2, -2);
```

Diese Nachricht ändert die Position der virtuellen Schallquelle mit der ID 1 auf die Koordinaten $x = 2$ und $y = -2$. Der Type Tag String ist bei WONDER nicht nötig.

3.4. Erweiterung von WONDER

Die Software WONDER bietet die Möglichkeit, virtuelle Schallquellen in ihrer Position zu verändern. Dabei beschränkt sich die Positionsänderung auf den zweidimensionalen Raum. In der aktuellen WONDER-Version ist es daher nur möglich, virtuelle Schallquellen entlang der x- und der y-Achse zu verschieben. Ziel der neuen Softwareversion soll es sein, virtuelle Schallquellen im dreidimensionalen Raum und somit auch in der Höhe entlang der z-Achse verschieben zu können.

3.4.1. Vorbereitung

Um WONDER für eine 3D-Quellenpositionierung zu erweitern, ist es nötig, ein Verständnis für den grundlegenden Aufbau und die Arbeitsweise der Software zu entwickeln. Die Software besteht aus mehreren Komponenten. Um einen allgemeinen Überblick über diese

¹<http://supercollider.sourceforge.net/>

Komponenten zu erhalten, bietet sich die mitgelieferte Dokumentation der Software an, in der jede Komponente grob beschrieben wird. Anhand dieser Dokumentation lässt sich feststellen, welche Komponente zur Steuerung der gesamten Software dient und wer das Rendering der virtuellen Schallquellen übernimmt. Da die Dokumentation jedoch auf keinerlei Details der Komponenten eingeht, ist eine Analyse des Quellcodes erforderlich.

Die für diese Arbeit wichtigen Komponenten beschränken sich auf `cwonder` und `twonder`. Laut Dokumentation stellt `cwonder` den Hauptteil der Software dar und kommuniziert mit jeder am System beteiligten Komponente. Die Kommunikation läuft innerhalb der Software über OSC-Nachrichten ab, wobei `cwonder` eintreffende Nachrichten verarbeitet und an die jeweiligen Komponenten weiterleitet. Die Kommunikation unter den Komponenten wird durch OSC-Server gewährleistet, wobei für jede Komponente ein OSC-Server gestartet wird und die für diese Komponente wichtigen Nachrichten definiert werden. Dabei stellt `twonder` die Rendering-Einheit dar und übernimmt die Berechnung der jeweiligen Audiosignale in Abhängigkeit von einer virtuellen Schallquelle. Hier findet die gesamte Verarbeitung einer virtuellen Schallquelle sowie die Verwaltung der am System teilnehmenden Lautsprecher statt. Aus der Laborkonfiguration geht hervor, dass `twonder` in mehreren Instanzen gestartet werden kann, wobei jede Instanz eine bestimmte Anzahl an Lautsprechern verwaltet [Chr14]. Innerhalb von Konfigurationsdateien wird festgelegt, welche Lautsprecher von einer Instanz verwaltet werden.

3.4.1.1. Cwonder

`Cwonder` gliedert sich in verschiedene Klassen, die auf mehrere Dateien ausgelagert sind. Den Hauptteil bildet dabei die Klasse `cwonder`, die die Softwarekomponente `cwonder` darstellt. Hier sind alle für den Betrieb notwendigen Funktionen und Variablen deklariert. `cwonder` wird von der Datei `main.cpp` aus gestartet, in der sich die ausführbare Main-Funktion befindet. Des Weiteren wird hier die Konfigurationsdatei für `cwonder` eingelesen, ein OSC-Server gestartet und `cwonder` letztendlich als Daemon ausgeführt und zur Verfügung gestellt.

`Cwonder` besitzt einen eigenen OSC-Server. Innerhalb der Datei `oscincrtl.cpp` befindet sich die Klasse `OSCControl`, in der bestimmte Nachrichtentypen definiert und dem Server bekannt gemacht werden. Für jede Nachricht existiert eine Callback-Funktion, die aufgerufen wird, sobald der Server die entsprechende Nachricht empfangen hat. Innerhalb dieser Funktion wird mit entsprechenden Argumenten eine Funktion aufgerufen, die sich innerhalb der Klasse `cwonder` befindet.

Damit teilnehmende Komponenten mit `cwonder` kommunizieren können, müssen diese sich einmalig bei `cwonder` registrieren. Für diesen Vorgang gibt es eigene Funktionen, mit denen sich

die jeweilige Komponente einem Stream anschließen kann. Diese Streams stellen nur diejenigen Informationen bereit, die für die am jeweiligen Stream angemeldeten Komponenten relevant sind [Han14]. Innerhalb der von den Callback-Funktionen aufgerufenen Funktionen entscheidet cwonder, an welchen Stream welche Informationen gesendet werden sollen. Insgesamt bietet cwonder vier solcher Streams an, jedoch sind für diese Arbeit nur zwei von Relevanz: der Visual Stream und der Render Stream. Über den Visual Stream werden alle Informationen gesendet, die für eine Visualisierung notwendig sind, über den Render Stream die Informationen, die für das Rendering entscheidend sind.

Abbildung 3.2 zeigt die schematische Darstellung der Kommunikation der Softwarekomponenten.

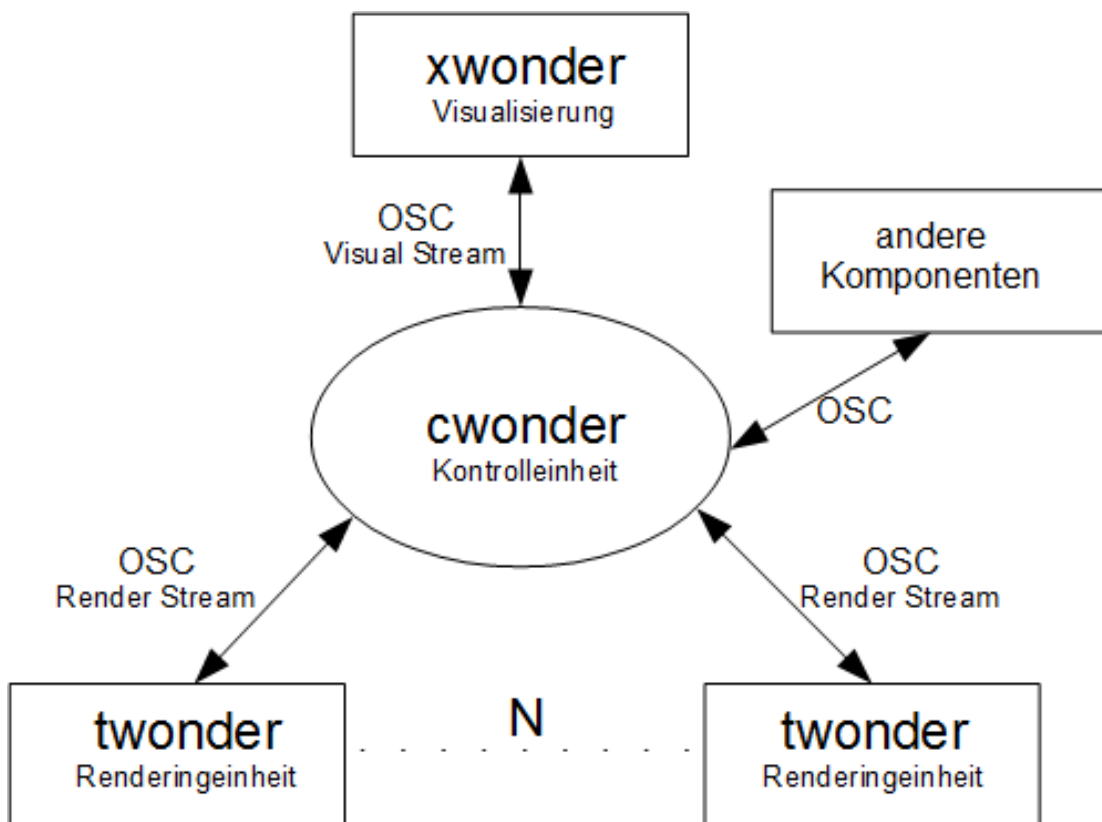


Abbildung 3.2.: Schematische Darstellung der Kommunikation der Softwarekomponenten

3.4.1.2. Twonder

Twonder übernimmt die Aufgabe des Renderings virtueller Schallquellen und gliedert sich, ähnlich wie cwonder, in mehrere Klassen. Die Datei twonder.cpp bildet den Hauptteil und

enthält die ausführbare Main-Funktion zum Starten der twonder-Komponente. Innerhalb dieser Main-Funktion wird die Konfigurationsdatei für twonder eingelesen und ein OSC-Server erstellt; die für diesen relevanten Nachrichten werden definiert und twonder wird als Daemon ausgeführt. Außerdem wird hier die Konfigurationsdatei der von twonder verwalteten Lautsprecher eingelesen.

Die jeweiligen Lautsprecher werden in der Klasse `speaker` angelegt, die sich innerhalb der Datei `speaker.cpp` befindet. Jedem Lautsprecher werden hier ein Name sowie seine Position und Ausrichtung als x-, y- und z-Koordinate zugewiesen. Die Daten der einzelnen Lautsprecher werden über die zugehörige Konfigurationsdatei innerhalb von `twonder.cpp` ausgelesen. Hier wird ein `Speakerarray` angelegt sowie ein `Speaker`-Objekt mit den jeweiligen Daten erstellt und dem `Speakerarray` hinzugefügt.

Eine virtuelle Schallquelle kann entweder als Punktquelle oder als Linearquelle erstellt werden. Die Punktquelle besitzt dabei eine Position, die durch eine x- und y-Koordinate festgelegt ist. Innerhalb der Klasse `source` sind die Funktionen für das Rendering implementiert. Aufgerufen werden diese von der Funktion `process` innerhalb der Datei `twonder.cpp`. Hier wird für jeden Lautsprecher abhängig von der virtuellen Schallquelle ein entsprechendes Signal berechnet.

Eine umfangreiche Beschreibung von twonder und des Renderings findet sich in der Bachelorarbeit von Carola Christoffel [Chr14].

3.4.1.3. Zweidimensionale Quellenpositionierung

Um die Position einer virtuellen Schallquelle innerhalb des zweidimensionalen Raumes zu verändern, werden drei verschiedene OSC-Nachrichten definiert. Diese drei Nachrichten unterscheiden sich dabei lediglich in der Anzahl der zu übertragenden Argumente. Nachfolgend sind diese drei Nachrichtentypen dargestellt.

```
"/WONDER/source/position", "iff", sourcePositionHandler  
"/WONDER/source/position", "iff", sourcePositionHandler  
"/WONDER/source/position", "iffff", sourcePositionHandler
```

Als erstes Argument wird die Adresse übergeben, die angibt, wohin die Nachricht gesendet werden soll. Als zweites Argument werden die Datentypen der zu übertragenden Argumente der Nachricht angegeben. Die Beschreibung richtet sich dabei nach dem Type Tag String des OSC-Protokolls. Als drittes Argument dient die Callback-Funktion, die beim Empfang der Nachricht aufgerufen wird. Die Adresse und die Callback-Funktion bleiben bei jeder Nachricht unverändert, jedoch verändert sich die Anzahl der zu übertragenden Argumente.

WONDER bietet die Möglichkeit, eine virtuelle Schallquelle über einen bestimmten Zeitraum von einer Position zu einer anderen laufen zu lassen sowie eine Nachricht zeitversetzt auszuführen. Die Daten, die für diese Operationen nötig sind, werden über diese Nachrichten verschickt und von der jeweiligen Callback-Funktion verarbeitet. So beinhaltet die erste Nachricht lediglich die Informationen für einen Positionswechsel der Schallquelle mit der ID und den x- und y-Koordinaten. Die zweite Nachricht erhält als zusätzlichen Wert eine Duration, die angibt, wie lange die Quelle benötigen soll, um zur nächsten Position zu gelangen. Die dritte Nachricht erhält neben der Duration einen Timestamp, der angibt, wann die Nachricht ausgeführt werden soll. Zu beachten ist hier, dass die Option zur Angabe eines Timestamps nur möglich ist, wenn auch eine Duration angegeben wurde, da sonst die Nachricht falsch interpretiert wird.

Innerhalb von `cwonder` besteht die Aufgabe der Callback-Funktion darin, die übergebenen Argumente zu verwalten und an die jeweilige Funktion innerhalb der Klasse `cwonder` weiterzuleiten. Diese Funktion sorgt dann für die Verteilung der Nachricht an die jeweiligen Streams. Des Weiteren sendet die Callback-Funktion eine Nachricht zurück an den Absender, die Informationen der aufgerufenen Funktion enthält. Hier wird dem Absender mitgeteilt, ob die Nachricht empfangen wurde oder ob es zu Fehlern bei der Ausführung kam.

Innerhalb von `twonder` haben die Callback-Funktionen eine ähnliche Aufgabe. Im Unterschied zu `cwonder` wird hier aber keine Funktion einer anderen Klasse aufgerufen, sondern die Aufgaben werden direkt in dieser Funktion durchgeführt. Außerdem werden dem Absender hier keine Informationen über den Ablauf einer Operation mitgeteilt.

3.4.1.4. Änderungen der Software

Mit dem nun vorhandenen Überblick über WONDER lassen sich grundlegende Schlussfolgerungen über die nötigen Änderungen innerhalb der Software ziehen. Da `twonder` das Rendering virtueller Schallquellen übernimmt, werden hier die für eine 3D-Quellenpositionierung nötigen Daten erwartet und verarbeitet. Die virtuellen Quellen, die aktuell im zweidimensionalen Raum positioniert werden können, benötigen eine weitere Koordinate, um die Berechnung im dreidimensionalen Raum durchzuführen. Des Weiteren muss eine kompatible Kommunikation mit den beteiligten Softwarekomponenten gewährleistet sein. Erreicht werden kann dies durch das Versenden von OSC-Nachrichten. Hierfür werden neue OSC-Nachrichten definiert und die dafür zuständigen Callback-Funktionen implementiert. Da außerdem zusätzliche Lautsprecher am System teilnehmen, muss eine neue Konfigurationsdatei erstellt und einer `twonder`-Instanz zugewiesen werden.

3.4.2. Implementierung der Änderungen

Basierend auf der Softwareversion von Carola Christoffel [Chr14], wird nachfolgend die Implementation der Änderungen von WONDER beschrieben.

3.4.2.1. Hinzufügen zusätzlicher OSC-Nachrichten

Die Kommunikation unter den einzelnen Komponenten findet über OSC-Nachrichten statt. Dabei besitzt jede Komponente einen OSC-Server, der zu Beginn gestartet wird und dem definierte Nachrichtentypen übergeben werden. Um eine 3D-Quellenpositionierung umzusetzen, muss für eindeutige Nachrichten innerhalb der Software gesorgt werden. Jede Softwarekomponente, die einen eigenen OSC-Server besitzt, leitet diesen von einer Wrapperklasse ab. Beim Start der jeweiligen Komponenten werden mit der Funktion `addMethod()` die jeweiligen OSC-Nachrichten dem Server übergeben. Folgendes Beispiel zeigt die Übergabe einer Nachricht an den Server.

```
addMethod( "/WONDER/source/position", "iff",  
          sourcePositionHandler, this );
```

Für die Positionierung einer Quelle im dreidimensionalen Raum werden drei neue Nachrichtentypen definiert. Diese Nachrichten besitzen einen ähnlichen Aufbau wie die bereits vorhandenen Nachrichten der Quellenpositionierung im zweidimensionalen Raum. Die Nachrichten bekommen die Adresse `/WONDER/source/position3D` und die Callback-Funktion `sourcePosition3DHandler`. Außerdem erhalten die zu übertragenden Argumente einen zusätzlichen Parameter vom Datentyp `Float` für die zusätzliche z-Koordinate.

```
"/WONDER/source/position3D", "iff", sourcePosition3DHandler  
"/WONDER/source/position3D", "iff", sourcePosition3DHandler  
"/WONDER/source/position3D", "iff", sourcePosition3DHandler
```

Die von `cwonder` verwalteten Nachrichten befinden sich in der Datei `oscincntrl.cpp`. Hier sind alle Nachrichten aufgelistet, die `cwonder` empfangen kann. Um die neu definierten Nachrichten `cwonder` bekannt zu machen, wird pro Nachricht jeweils ein Funktionsaufruf von `addMethod` mit der jeweiligen Nachricht hinzugefügt. Als Beispiel folgt der Methodenaufruf für die erste Nachricht.

```
addMethod( "/WONDER/source/position3D", "iff",  
          sourcePosition3DHandler, this );
```


Mit demselben Vorgehen werden die drei Nachrichten innerhalb von `twonder` definiert. Im Unterschied zu `cwonder` besitzt `twonder` keine ausgelagerte Datei zur Definition von OSC-Nachrichten für den OSC-Server. Hier werden die Nachrichten direkt in der Main-Funktion definiert. Der Aufbau der Nachrichten ändert sich dabei nur geringfügig. Die Callback-Funktion bekommt hier die Bezeichnung `oscSrcPositionHandler3D`.

3.4.2.2. Hinzufügen zusätzlicher Callback-Funktionen

Mit der Definition der zusätzlichen Nachrichten muss ebenfalls eine zusätzliche Callback-Funktion implementiert werden. Diese Callback-Funktion wird beim Empfang einer der drei Nachrichten aufgerufen. Der Aufbau dieser Funktion hält sich dabei an die Struktur der Callback-Funktion der zweidimensionalen Quellenpositionierung.

Die Erweiterung von `cwonder` gliedert sich in zwei Schritte. Als Erstes wird in der Klasse `cwonder` eine neue Funktion mit der Bezeichnung `setSourcePosition3D` erstellt. Diese Funktion wird von der Callback-Funktion beim Empfang einer der neu definierten Nachrichten aufgerufen und erhält einen zusätzlichen Parameter `float z`, der die neue z-Koordinate darstellt. Aufgabe dieser Funktion ist es, die Informationen der erhaltenen Nachricht an die jeweiligen Streams zu senden. Die hier ausschlaggebenden Streams sind der Render Stream und der Visual Stream. Da bis zu diesem Zeitpunkt keine 3D-fähige Komponente besteht, werden über den Visual Stream nach wie vor nur x- und y-Koordinaten an die ursprüngliche Adresse verschickt. Beim Render Stream hingegen wird eine Nachricht an die neu erstellte Adresse `/WONDER/source/position3D` geschickt mit dem zusätzlichen Argument `z` zur Angabe der z-Koordinate.

Anschließend wird in der Datei `oscincntrl.cpp` eine neue Callback-Funktion mit der Bezeichnung `sourcePosition3DHandler` erstellt. Der einzige Unterschied zur Callback-Funktion der zweidimensionalen Quellenpositionierung besteht darin, einen zusätzlichen Wert zu verarbeiten. Da die Nachrichten nun ebenfalls eine z-Koordinate übertragen, muss die Verarbeitung der erhaltenen Daten angepasst werden. Für die Verarbeitung des optional enthaltenen Duration- und Timestamp-Arguments reicht eine einfache Erhöhung der Vergleichsparameter und der Indizes der Argumente. Um die erhaltenen Informationen an die zuvor erstellte Funktion innerhalb der Klasse `cwonder` zu übergeben, wird ein üblicher Funktionsaufruf mit den jeweiligen Argumenten vollzogen.

Die Erweiterung von `twonder` gestaltet sich ähnlich wie die von `cwonder`. Innerhalb der Datei `twonder.cpp` wird eine neue Callback-Funktion mit der Bezeichnung `oscSrcPositionHandler3D` erstellt. Der Aufbau dieser Funktion gestaltet sich ebenfalls ähnlich wie die der zweidimensionalen Quellenpositionierung, allerdings mit dem Unterschied der zusätzlich zu verarbeitenden

z-Koordinate. Um diese in den Programmablauf zu integrieren, wird eine zusätzliche Variable mit der Bezeichnung `newZ` erstellt. Diese Variable stellt im weiteren Programmfluss die z-Koordinate dar. Des Weiteren muss auch hier das optional enthaltene Duration- und Timestamp-Argument verarbeitet werden. Dies geschieht analog zum Vorgehen in `cwonder`. Abbildung 3.3 zeigt die hinzugefügten Callback-Funktionen innerhalb der Komponenten und den Ablauf beim Empfang einer Nachricht zur Quellenpositionierung.

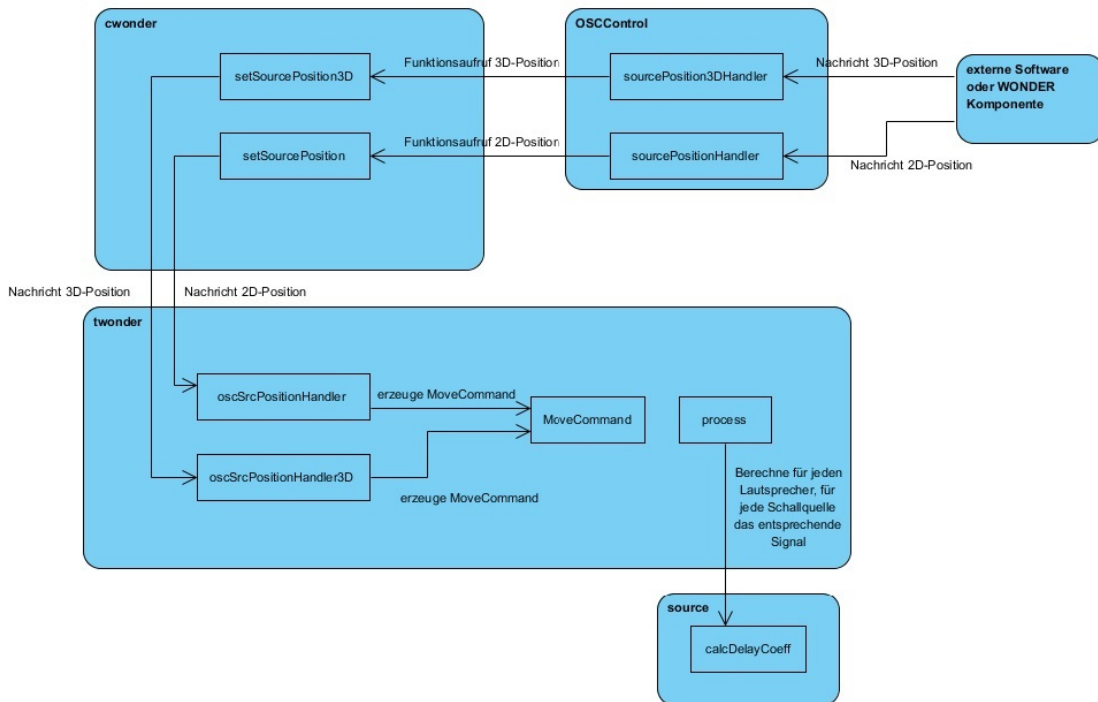


Abbildung 3.3.: Ablauf nach Hinzufügen zusätzlicher Callback-Funktionen

3.4.2.3. Anpassen der Datentypen

Virtuelle Schallquellen besitzen eine Position, die sich in eine x- und eine y-Koordinate gliedert. Da in der erweiterten WONDER-Version die Schallquellen zusätzlich entlang der z-Koordinate positioniert werden sollen, müssen die Positionen entsprechend angepasst werden. Bei Linearquellen ist darauf zu achten, ebenfalls den Winkel mit anzupassen, da Linearquellen neben einer Ursprungsposition auch eine Richtung vorweisen.

Die Position einer virtuellen Schallquelle wird dargestellt durch den Datentyp `Vector2D`. Dieser Datentyp enthält zwei Variablen, die die Position der Schallquelle in x- und y-Richtung angeben. Um die Schallquellen durch eine dritte Koordinate darstellen zu können, muss ein

neuer Datentyp erstellt werden. Innerhalb der WONDER-Software ist ein derartiger Datentyp bereits vorhanden. `Vector3D` hat eine ähnliche Struktur zu `Vector2D`, jedoch mit einer dritten Variable, die die z-Koordinate enthält. Dieser Datentyp wird genutzt, um die virtuelle Schallquelle in der dreidimensionalen Ebene positionieren zu können.

Um dieses Ziel zu erreichen, werden sämtliche `Vector2D`-Datentypen in `Vector3D`-Datentypen geändert. Die dafür nötigen Anpassungen werden in den Dateien `source.h`, `source.cpp` und `t wonder.cpp` durchgeführt. Innerhalb der Datei `source.h` ist ein einfaches Austauschen von `Vector2D` durch `Vector3D` möglich, jedoch ist hier darauf zu achten, dass sämtliche Parameter der definierten Funktionen mit angepasst werden. Die Änderungen der Datei `source.cpp` lassen sich ähnlich durchführen, jedoch ist hier darauf zu achten, dass innerhalb der Funktion `calcDelayCoeff` anstatt der zweidimensionalen Position der Lautsprecher die dreidimensionale Position abgefragt wird. Die nötigen Funktionen dafür werden von der Software standardmäßig bereitgestellt. Innerhalb der Datei `t wonder.cpp` wird wie zuvor der Datentyp `Vector2D` durch `Vector3D` ersetzt und die definierten Funktionen dementsprechend angepasst.

3.4.2.4. Anpassen der Konfiguration

Für die Positionierung virtueller Schallquellen im dreidimensionalen Raum werden vier zusätzliche Lautsprecher in das System eingebunden. Das Hinzufügen dieser Lautsprecher in das technische System wurde bereits in Abschnitt 2.5.2 beschrieben.

Für die Verwaltung der Lautsprecher innerhalb von WONDER ist die Komponente `t wonder` zuständig. Über Konfigurationsdateien im XML-Format werden die Daten der Lautsprecher beim Starten von `t wonder` eingelesen. Jede Konfigurationsdatei enthält bestimmte Lautsprecher und wird von einer bestimmten `t wonder`-Instanz eingelesen. Die Struktur der Konfigurationsdateien gliedert sich dabei in eine Lautsprecherkomponente pro Zeile. Hier werden die ID, die Anzahl an Wiedergabekanälen sowie die Position und die Ausrichtung der Lautsprecher angegeben.

Um die zusätzlichen Lautsprecher softwareseitig nutzbar zu machen, wird eine neue Konfigurationsdatei mit der Bezeichnung `t wonder_speakerarray8.xml` erstellt. Hier werden die zusätzlichen Lautsprecher zeilenweise mit deren spezifischen Daten eingetragen. Die Lautsprecher werden an der kurzen Seite der Wellenfeldsynthese-Anlage positioniert, an der sich auch die Powerwall befindet. Diese Lautsprecher bilden die untere Lautsprecherebene der Wellenfeldsynthese-Anlage, daher wurde bei der Positionierung der Lautsprecher darauf geachtet, dass sie sich gleichmäßig unter der oberen Lautsprecherebene verteilen. Jeder untere Lautsprecher befindet sich demnach mittig unter einem oberen Lautsprechersegment. Somit lassen

sich aus den Koordinaten der oberen Lautsprechersegmente die Koordinaten $[-1.2230, 0, 0.2]$, $[-0.4077, 0, 0.2]$, $[0.4077, 0, 0.2]$ und $[1.2230, 0, 0.2]$ für die unteren Lautsprecher berechnen. Für die Verwaltung der zusätzlichen Lautsprecher wird eine neue twonder-Instanz genutzt. Um dem System mitzuteilen, dass eine zusätzliche Instanz gestartet werden soll, wird in der Datei `twonder_n101.conf`, die sich unter den Startskripten finden lässt, der Wert 4 hinzugefügt. Der Wert 4 gibt dabei die zu verwaltenden Wiedergabekanäle an. Abbildung 3.4 stellt die kurze Seite der Wellenfeldsynthese-Anlage mit Zusatzlautsprechern dar. Anhand des Koordinatenkreuzes und der eingetragenen Koordinaten der Lautsprechermodule und Zusatzlautsprecher lässt sich die Aufstellung und gleichmäßige Verteilung der unteren Lautsprecher gut erkennen.

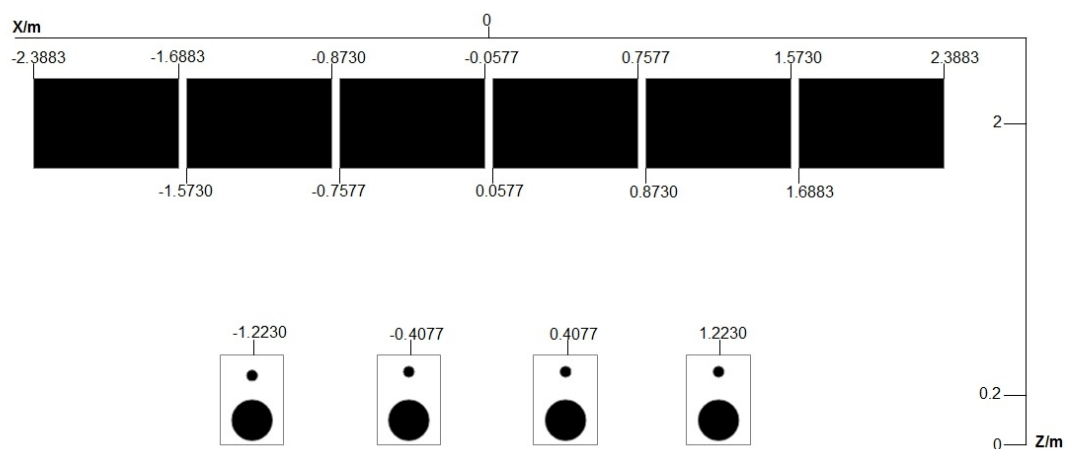


Abbildung 3.4.: Erweiterung durch Zusatzlautsprecher. Koordinaten der oberhalb gelegenen sechs Lautsprechermodule mit jeweils acht Wiedergabekanälen und der vier unterhalb gelegenen zusätzlichen Lautsprecher

Um die zusätzlichen Lautsprecher bestmöglich in das bestehende System einzufügen, wurde mittels eines mittig zwischen den Lautsprecherebenen positionierten Studiomikrofons und der Digital Audio Workstation Cubase die Lautstärke der zusätzlichen Lautsprecher an das bestehende System angepasst. Dazu wurde jeweils ein Testsignal auf der Position eines oberen Wiedergabekanals und auf der Position eines unteren Wiedergabekanals gleichzeitig abgespielt und aufgenommen. Cubase bietet die Möglichkeit, die Lautstärke der aufgenommenen Signale auszuwerten. Durch den analogen Regler an der Front der zusätzlichen Lautsprecher wurde im Anschluss die Lautstärke durch mehrere Messungen an das System angepasst. Eine Korrektur der Laufzeit zwischen den unterschiedlichen Lautsprechern wurde nicht berücksichtigt.

3.4.2.5. Kompatibilität

Eine Positionierung virtueller Schallquellen im dreidimensionalen Raum soll lediglich als Erweiterung der bestehenden Software betrachtet werden. Nach wie vor soll es möglich sein, das System auf herkömmliche Weise zu nutzen. So soll es weiterhin gewährleistet sein, Schallquellen im zweidimensionalen Raum positionieren und sämtliche bestehenden Funktionen uneingeschränkt ausführen zu können.

Garantiert wird dies durch die Erweiterung der Software. Alle zuvor enthaltenen Funktionen wurden erhalten, jedoch an gewissen Stellen für eine kompatible Ausführung angepasst. Das Hauptaugenmerk lag dabei auf der gleichzeitigen Nutzung einer 2D- und einer 3D-Quellenpositionierung. Innerhalb der Software kann eine virtuelle Schallquelle nur als zweidimensionales Objekt, also mit x- und y- Koordinate, oder als dreidimensionales Objekt, mit x-, y- und z-Koordinate, bestehen. Durch die Erweiterung einer 3D-Quellenpositionierung besteht eine Schallquelle als dreidimensionales Objekt. Wird für ein solches Objekt eine Nachricht empfangen, die nur aus einer x- und y-Koordinate besteht, kann diese von twonder nicht verarbeitet werden. Damit dies jedoch weiterhin möglich ist, wird ein Standardwert innerhalb der Callback-Funktion `oscSrcPositionHandler` implementiert. Dieser Standardwert stellt die fehlende z-Koordinate dar. Wird eine entsprechende Nachricht empfangen, wird der Standardwert als z-Koordinate interpretiert und ermöglicht somit eine korrekte Verarbeitung. Da die obere Lautsprecherebene auf zwei Metern Höhe liegt und sich eine zweidimensionale Quellenpositionierung nicht auf die Höhe einer Schallquelle bezieht, wird dem Standardwert der Wert 2 zugewiesen.

WONDER bietet die Möglichkeit, Projekte zu erstellen und diese zu verwalten. Da durch die Erweiterung einer 3D-Quellenpositionierung eine virtuelle Schallquelle als dreidimensionales Objekt besteht, in gespeicherten Projekten der vorigen Version jedoch als zweidimensionales Objekt gespeichert wurde, kann das Projekt durch Fehlen der z-Koordinate nicht korrekt geladen werden. Um dies gewährleisten zu können, sind zwei Änderungen erforderlich. Zunächst wird die Datei `cwonder_project.dtd` um die zusätzliche Zeile `posz CDATA #IMPLIED` erweitert. Dabei stellt `posz` das Attribut und somit die z-Koordinate und `CDATA` den Datentyp dar, während `#IMPLIED` das Attribut als optional markiert. Optional bedeutet hier, dass beim Laden des Projekts die jeweilige virtuelle Schallquelle auf das Vorhandensein einer z-Koordinate geprüft wird. Anschließend wird der Software im Rahmen der Funktion `readFromDOM` der Datei `project.cpp` mitgeteilt, dass bei Fehlen einer z-Koordinate diese auf einen Standardwert gesetzt wird. Wird nun ein Projekt der vorigen Softwareversion geladen, registriert die Software das Fehlen der z-Koordinate und reagiert entsprechend.

3.4.2.6. Einschränkung der Nutzung

Beim Ausführen der Software kann es Einschränkungen der am System beteiligten Anwendungen geben. Wird das System ausschließlich für die Positionierung virtueller Schallquellen in der zweidimensionalen Ebene genutzt, ist eine uneingeschränkte Ausführung möglich. Wird jedoch während des Betriebs zwischen zweidimensionaler und dreidimensionaler Ebene gewechselt, kann es zu Konflikten mit den teilnehmenden Anwendungen kommen. Ursache hierfür ist die Verarbeitung von OSC-Nachrichten innerhalb dieser Anwendungen. Wurden diese für den Betrieb in zweidimensionaler Ebene entworfen, werden dementsprechend keine Nachrichten verarbeitet, die bei einer 3D-Quellenpositionierung von der Software verschickt werden. Das hat zur Folge, dass die Schallquelle von dieser Anwendung nicht verarbeitet werden kann, was wiederum zu eventuellen Fehlfunktionen führt.

3.4.3. Ergebnis

Das Ergebnis ist eine Version der Software WONDER, in der es möglich ist, virtuelle Schallquellen in der dreidimensionalen Ebene zu positionieren. Hierfür wurden innerhalb der Software neue OSC-Nachrichten und Funktionen implementiert, die die Übertragung einer zusätzlichen Koordinate möglich machen. Da dies lediglich als Erweiterung der bestehenden Softwareversion dient, ist eine uneingeschränkte Nutzung der Software bei einer zweidimensionalen Quellenpositionierung möglich. Der gemischte Betrieb zwischen einer zwei- und dreidimensionalen Quellenpositionierung kann jedoch zu Problemen bei zusätzlichen Anwendungen führen.

4. Simulation des Schallfeldes

Um die durch die Wellenfeldsynthese synthetisierten Wellenfronten analysieren zu können, wurde mit der Software MATLAB¹ von der Firma The MathWorks Inc.² ein Verfahren entwickelt, das die durch Schallquellen erzeugten Schallfelder anhand der entstehenden Intensitäten grafisch darstellt. Dabei wird der für diese Arbeit ausschlaggebende Bereich der Wellenfeldsynthese-Anlage simuliert, der sich auf die kurze Seite der Anlage beschränkt, an der sich die Zusatzauslautsprecher befinden. Insgesamt werden 52 Wiedergabekanäle simuliert, die sich aus sechs Lautsprechermodulen auf der Höhe von 2 m und vier Lautsprechern auf der Höhe von 0,2 m zusammensetzen.

4.1. Aufbau

Der Programmablauf stellt sich wie in Abbildung 4.1 gezeigt dar. Für die Simulation eines Schallfeldes ist es nötig, dem Programm die nötigen Informationen über die Lage der Wiedergabekanäle und des Schallfeldes zu übermitteln. Hierzu sind einige Variablen mit festen Werten zu initialisieren, die die Koordinaten des ersten Lautsprechermoduls sowie die Start- und Endposition des zu berechnenden Schallfeldes darstellen. Außerdem können für jede Berechnung bestimmte Variablen mit individuellen Werten belegt werden. So kann festgelegt werden, auf welcher Position sich die virtuelle Schallquelle befindet, in welchen Abständen sich die Messpositionen befinden und wie viele Wiedergabekanäle am System teilnehmen. Basierend auf den Koordinaten der Start- und Endposition des zu berechnenden Schallfeldes, wird ein Array mit Messpunkten erstellt. Ausgehend von dem ersten Messpunkt, der die Startkoordinate des Schallfeldes darstellt, wird in x-Richtung jeweils ein weiterer Messpunkt in dem vorgegebenen Messabstand hinzugefügt. Wird die Endposition des Schallfeldes erreicht, wird die x-Koordinate auf den Startpunkt zurückgesetzt und die y-Koordinate um den zuvor festgelegten Abstand erhöht.

Für die am System teilnehmenden Wiedergabekanäle wird ein separates Array angelegt. Dieses Array enthält die Koordinaten eines jeden am System teilnehmenden Wiedergabekanal.

¹<http://mathworks.com/products/matlab/>

²<http://www.mathworks.com/>

Als Startpunkt der Berechnung dienen die Koordinaten des äußeren links oben gelegenen Lautsprechermoduls. Für die Lautsprechermodule gilt ein einheitlicher Abstand von 10 cm zwischen den Wiedergabekanälen, jedoch mit einer Besonderheit: Zwischen den einzelnen Lautsprechermodulen erhöht sich der Abstand auf etwa 11 cm. Die unteren Lautsprecher besitzen jeweils einen Wiedergabekanal mit einem Abstand von etwa 80 cm. Durch Addition des Abstandes zu der Koordinate des vorherigen Wiedergabekanal wird die Koordinate des nächsten Wiedergabekanal berechnet und dem Array hinzugefügt.

Um das Schallfeld korrekt zu simulieren, sind zwei elementare Schritte nötig. Als Erstes muss für jeden Wiedergabekanal ein resultierendes Audiosignal, das mittels komplexer Exponentialfunktion dargestellt wird, berechnet werden. Dieses Audiosignal setzt sich aus einer Amplitude a_i und einem Phasenwinkel ϕ_i zusammen. Die Amplitude wird dabei durch den vorgegebenen Algorithmus der WONDER-Software berechnet, der sich in der Datei `source.cpp` befindet. Dieser Algorithmus berechnet per Koordinaten der Schallquelle und der Wiedergabekanal das resultierende Audiosignal jedes Wiedergabekanal. Eine detaillierte Darstellung des Algorithmus und eine Herleitung der Formel zur Berechnung der Amplitude bietet die Dissertation von Marije Baalman [Baa07]. Zur Berechnung des Phasenwinkels ϕ wird der Abstand des Wiedergabekanal zur virtuellen Schallquelle und die Wellenlänge, in Folgendem als l und λ dargestellt, benötigt. Die Wellenlänge λ wird durch

$$\lambda = \frac{c}{f} \quad (4.1)$$

berechnet, wobei c die Schallgeschwindigkeit und f die Frequenz ist. Die Länge l steht zum Phasenwinkel ϕ in folgender Beziehung [Föl]:

$$\frac{l}{\lambda} = \frac{\phi}{2\pi} \quad (4.2)$$

Umgeformt ergibt sich:

$$\phi = 2\pi \frac{l}{\lambda} \quad (4.3)$$

Daraus resultierend lässt sich das Audiosignal eines Wiedergabekanal folgendermaßen darstellen:

$$a_i e^{j\phi_i} \quad (4.4)$$

Im Anschluss daran wird ein von der Distanz des Messpunktes zum Wiedergabekanal abhängiges Signal an der jeweiligen Messpositionen berechnet. Dazu wird der Abstand jedes

4. Simulation des Schallfeldes

Wiedergabekanals zum Messpunkt, im Folgendem d_i genannt, genommen und daraus die resultierende Amplitude und der Phasenwinkel ϕ_{d_i} berechnet. Die Amplitude ergibt sich dabei aus dem Abstandsgesetz, das besagt, dass die Amplitude mit $\frac{1}{d}$ der Distanz einer punktförmigen Schallquelle abnimmt[Sen].

$$\phi_{d_i} = 2\pi \frac{d_i}{\lambda} \quad (4.5)$$

Daraus ergibt sich folgendes Signal:

$$\frac{1}{d_i} e^{j\phi_{d_i}} \quad (4.6)$$

Das Ergebnissignal an einem Messpunkt lässt sich durch Multiplikation des distanzabhängigen Signals am Messpunkt mit dem Signal des Wiedergabekanals und die anschließende Aufsummierung der Signale aller beteiligten Wiedergabekanäle berechnen:

$$s_{mp_i} = \sum_{i=1}^N \frac{a_i}{d_i} e^{j(\phi_i + \phi_{d_i})} \quad (4.7)$$

Um die letztendlich resultierenden Schallintensitäten darzustellen, werden die Beträge der komplexen Zahlen quadriert und mittels eines 2D-Plots dargestellt.

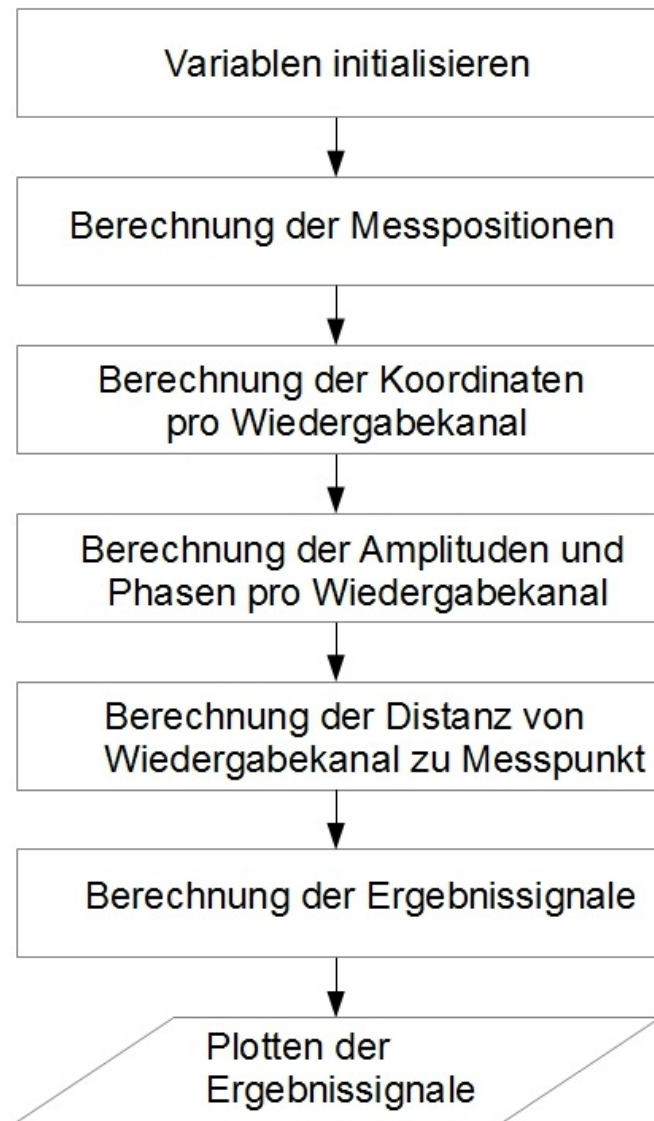


Abbildung 4.1.: Ablauf zur Berechnung der Schallintensitäten eines Schallfeldes

4.2. Resultate

Basierend auf dem in Abschnitt 4.1 beschriebenen Simulationsverfahren, werden in diesem Abschnitt simulierte Schallfelder anhand bestimmter Quellenpositionen dargestellt. Die Grafiken halten sich dabei jeweils an ein bestimmtes Muster, wobei die x- und die y-Achse den Bereich des simulierten Schallfeldes darstellen und die farblichen Verläufe die Intensitätswerte angeben. Anzumerken ist, dass es sich um zweidimensionale Grafiken handelt, jedoch bei der Beschreibung der Quellenpositionierung der Höhenbereich in Form einer z-Koordinate mit angegeben wird. Die Höhe der Messpunkte befindet sich dabei immer auf einer Ebene mit der Schallquelle. Die in nachfolgenden Simulationen simulierten Schallquellen besitzen jeweils ein Signal mit einer Frequenz von 500 Hz.

4.2.1. Simulation der oberen Lautsprecherebene

Um einen grundlegenden Eindruck von der Verteilung der Schallintensitäten zu bekommen, wird eine Schallquelle mit den Koordinaten $[0, -0.1, 2]$ simuliert, die dementsprechend mittig und 0,1 m hinter der oberen Lautsprecherebene platziert ist. Die Simulation findet hier ohne die Erweiterung der zusätzlichen Lautsprecher statt und bezieht sich nur auf die oberen 48 Wiedergabekanäle. Zu beachten ist hier, dass die z-Koordinate der Schallquelle den festen Wert 2 haben muss.

Die Abbildungen 4.2 und 4.3 stellen die Verteilungen der Intensitäten bei einer Quellenpositionierung mit den Koordinaten $[0, -0.1, 2]$ und $[0, -1, 2]$ dar. Abbildung 4.2 zeigt eine sich gleichmäßig ausbreitende Verteilung der Intensitäten, ausgehend von der Position der Schallquelle. Diese Art der Entwicklung erinnert stark an eine sich ausbreitende Wellenfront, wie sie in Abbildung 2.1 dargestellt ist. Im Vergleich dazu ergibt sich in Abbildung 4.3 keine gleichmäßige Verteilung der Intensitäten. Es ist zwar eine ähnliche Ausbreitung zu erkennen, jedoch wird diese von mehreren Unebenheiten durchzogen.

4.2.2. Hinzufügen der Zusatzlautsprecher

Nachfolgend werden die Zusatzlautsprecher in die Simulation mit einbezogen. Das simulierte System besteht somit aus 52 Wiedergabekanälen, wobei 48 Wiedergabekanäle auf der Höhe von 2 m liegen und weitere vier auf der Höhe von 0,2 m.

4. Simulation des Schallfeldes

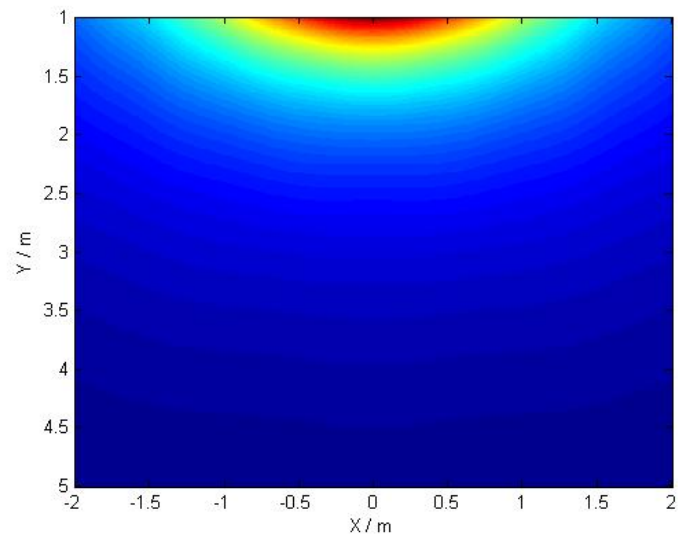


Abbildung 4.2.: Schallintensität einer Schallquelle mit Koordinaten $[0, -0.1, 2]$, berechnet nach Gleichung 3.7; Lautsprecher auf Höhe von 2 m und mit Abstand von 0,1 m; Frequenz 500 Hz

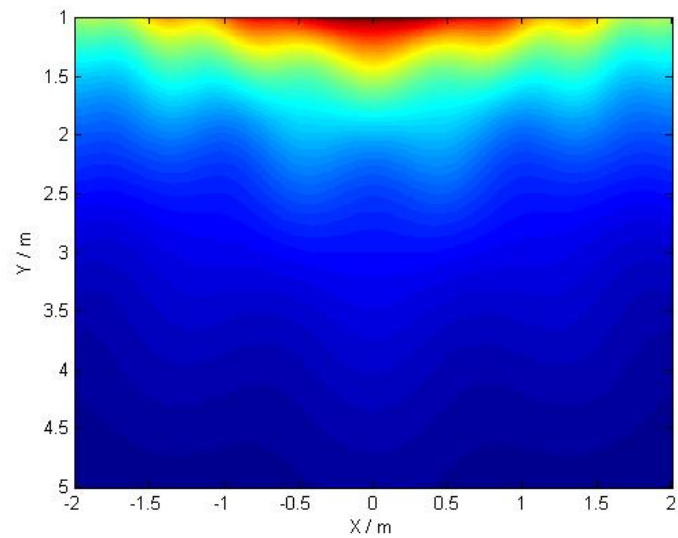


Abbildung 4.3.: Schallintensität einer Schallquelle mit Koordinaten $[0, -1, 2]$, berechnet nach Gleichung 3.7; Lautsprecher auf Höhe von 2 m und mit Abstand von 0,1 m; Frequenz 500 Hz

4. Simulation des Schallfeldes

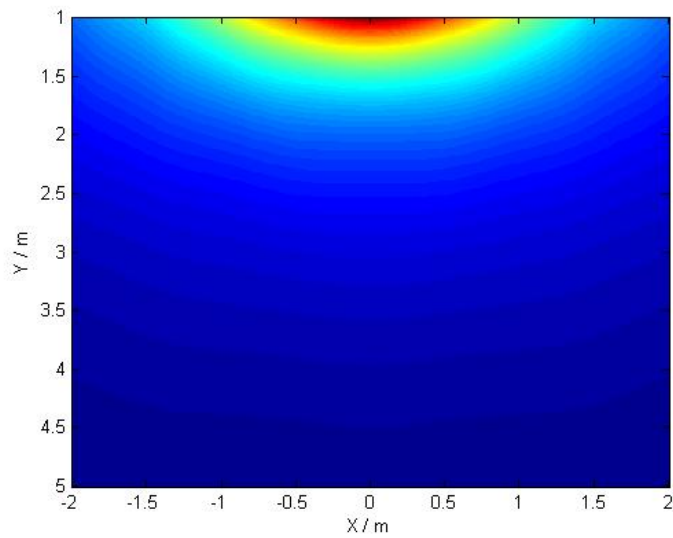


Abbildung 4.4.: Schallintensität einer Schallquelle mit Koordinaten $[0, -0.1, 2]$, berechnet nach Gleichung 3.7; Lautsprecher auf Höhe von 2 m und mit Abstand von 0,1 m sowie Zusatzlautsprecher auf Höhe von 0,2 m und mit Abstand von 0,8 m; Frequenz 500 Hz

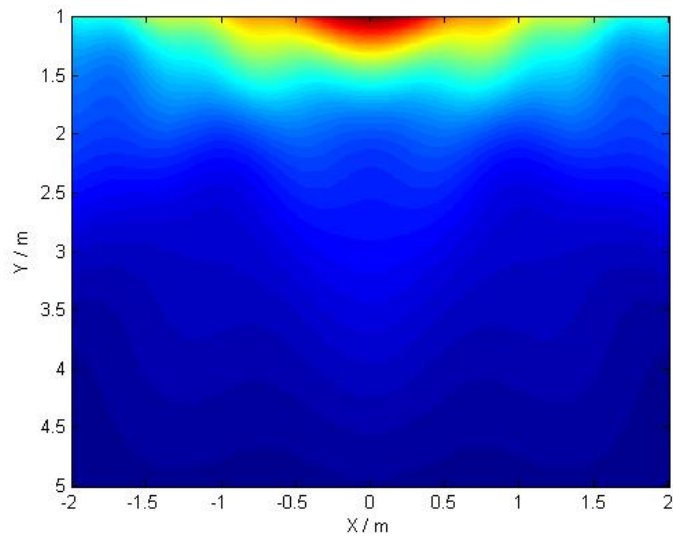


Abbildung 4.5.: Schallintensität einer Schallquelle mit Koordinaten $[0, -1, 2]$, berechnet nach Gleichung 3.7; Lautsprecher auf Höhe von 2 m und mit Abstand von 0,1 m sowie Zusatzlautsprecher auf Höhe von 0,2 m und Abstand von 0,8 m; Frequenz 500 Hz

Die Abbildungen 4.4 und 4.5 stellen die Verteilungen der Intensitäten bei derselben Quellenpositionierung der Abbildungen 4.2 und 4.3 dar. Wird Abbildung 4.2 im Vergleich zu 4.4 betrachtet, ist eine identische Ausbreitung der Intensitäten zu beobachten. Werden hingegen die Abbildungen 4.3 und 4.5 verglichen, lassen sich geringfügige Unterschiede feststellen. So ist zu erkennen, dass die Verteilung der Intensität bei Verwendung der Zusatzlautsprecher zu den Rändern eher abfällt, als es ohne die Zusatzlautsprecher der Fall ist. Der unebene Verlauf bleibt jedoch bestehen. Im direkten Vergleich der Systeme mit 48 bzw. 52 Wiedergabekanälen und bei gleicher Quellenpositionierung zeigt sich, dass der Einfluss der Signale der zusätzlichen Lautsprecher von der Positionierung der simulierten Schallquelle abhängt.

4.2.3. Höhenveränderung der Schallquelle

Durch das Hinzufügen der zusätzlichen Lautsprecher lässt sich die Position einer Schallquelle zusätzlich in der Höhe verändern.

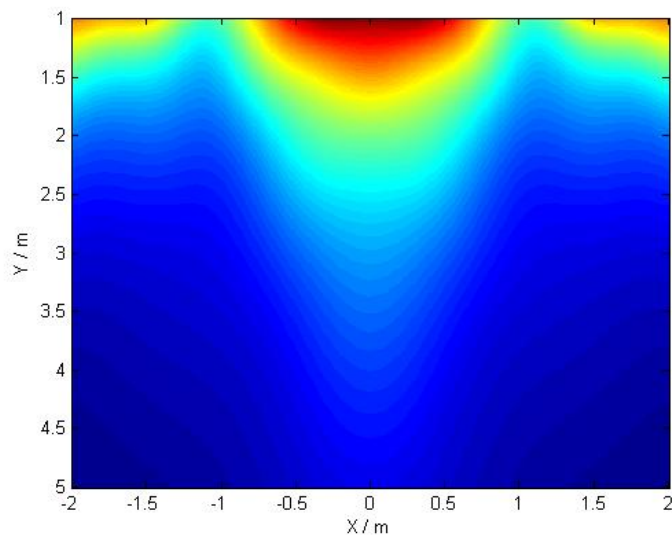


Abbildung 4.6.: Schallintensität einer Schallquelle mit Koordinaten $[0, -0.1, 1.1]$, berechnet nach Gleichung 3.7; Lautsprecher auf Höhe von 2 m und mit Abstand von 0,1 m sowie Zusatzlautsprecher auf Höhe von 0,2 m und Abstand von 0,8 m; Frequenz 500 Hz

Abbildung 4.6 zeigt eine Schallquelle mit den Koordinaten $[0, -0.1, 1.1]$. Die Schallquelle befindet sich hier nicht mehr auf 2 m Höhe, wie es zuvor der Fall gewesen ist, sondern exakt mittig zwischen der oberen und unteren Lautsprecherebene. Betrachtet man die entsprechende Verteilung der Intensitäten im Vergleich zu den vorherigen Abbildungen, lassen sich deutliche

4. Simulation des Schallfeldes

Unterschiede erkennen. So ist hier keine gleichmäßige Ausbreitung zu beobachten, sondern eine von den Kanten her abfallende Intensität, die auf der Höhe der Schallquelle wieder ansteigt. Auffallend ist, dass der Verlauf auf der Höhe der äußeren unteren Lautsprecher deutliche Einbuchtungen aufweist, die Intensität auf der Höhe der inneren Lautsprecher aber wieder stark ansteigt.

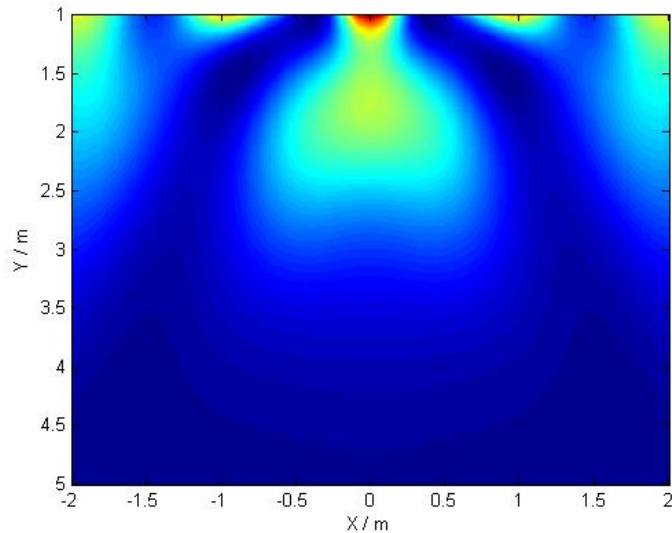


Abbildung 4.7.: Schallintensität einer Schallquelle mit Koordinaten $[0, -0.1, 0.2]$, berechnet nach Gleichung 3.7; Lautsprecher auf Höhe von 2 m und mit Abstand von 0,1 m sowie Zusatzlautsprecher auf Höhe von 0,2 m und mit Abstand von 0,8 m; Frequenz 500 Hz

Abbildung 4.7 zeigt eine Schallquelle mit den Koordinaten $[0, -0.1, 0.2]$. Diese Schallquelle befindet sich somit auf der Höhe der unteren Lautsprecherebene. Wird der direkte Vergleich mit einer auf der Höhe der oberen Lautsprecherebene gelegenen Schallquelle, wie in Abbildung 4.4 dargestellt, gezogen, lassen sich deutliche Unterschiede feststellen. Abbildung 4.7 zeigt eine im Vergleich ungleichmäßige Verteilung der Intensitäten. Auffallend sind hier die ansteigenden Intensitäten am Rand und auf der Höhe der unteren Lautsprecher sowie die abfallende Intensität zwischen diesen Lautsprechern.

Abbildung 4.8 zeigt eine Schallquelle mit den Koordinaten $[0, -0.1, 0.2]$, wobei ausschließlich die am System teilnehmenden Zusatzlautsprecher simuliert werden. Wird Abbildung 4.7 mit dieser verglichen, sind deutlich die Einflüsse der Wiedergabekanäle der oberen Lautsprecherebene erkennbar. Werden diese jedoch vom System ausgeschlossen, entsteht eine Verteilung der Intensitäten, die sich auf den Bereich um die Positionen der Zusatzlautsprecher begrenzt.

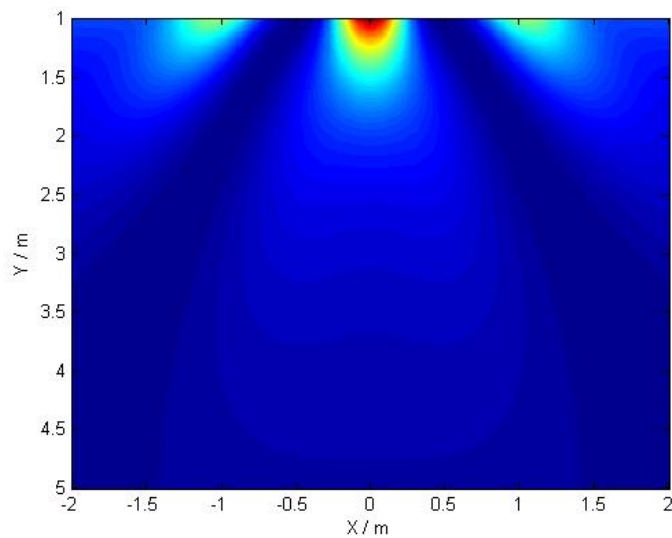


Abbildung 4.8.: Schallintensität einer Schallquelle mit Koordinaten $[0, -0.1, 0.2]$, berechnet nach Gleichung 3.7; Zusatzlautsprecher auf Höhe von 0,2 m und mit Abstand von 0,8 m; nur Zusatzlautsprecher aktiv; Frequenz 500 Hz

Deutlich zu erkennen ist dies durch die sich bildenden Maxima auf der Höhe der Positionen der Zusatzlautsprecher, die sich an den x-Positionen $-1.2230, -0.4077, 0.4077$ und 1.2230 befinden.

4.2.4. Positionsabhängigkeit des Hörers

Abbildung 4.9 nimmt Bezug auf verschiedene Hörpositionen mit dem Ziel, die Verteilung der Intensität bei einer sich wechselnden Höhe der Hörposition darzustellen und so den Einfluss der Positionsänderung in vertikaler Richtung zu ergründen. Dargestellt sind drei simulierte Schallfelder mit einer Schallquelle, die jeweils an den Koordinaten $[0, -0.1, 1.1]$ positioniert ist.

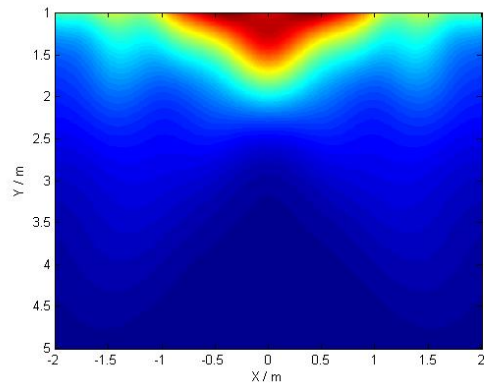
Schallfeld (a) zeigt die Verteilung der Intensitäten bei einer Hörposition von 1,7 m Höhe. Deutlich zu erkennen ist ein leicht unebener Anstieg der Intensität zur Position der Schallquelle hin, wobei das Maximum auf der Höhe der Schallquelle liegt. Außerdem lässt sich eine leicht erhöhte Intensität an den Positionen der äußeren unteren Lautsprecher erkennen.

Schallfeld (b) zeigt die Verteilung der Intensitäten bei einer Hörposition von 1,1 m Höhe. Die Hörposition befindet sich hier auf einer Höhe mit der Schallquelle. Zu erkennen ist ähnlich wie bei Schallfeld (a) ein Anstieg der Intensität zur Position der Schallquelle hin. Auf der Höhe

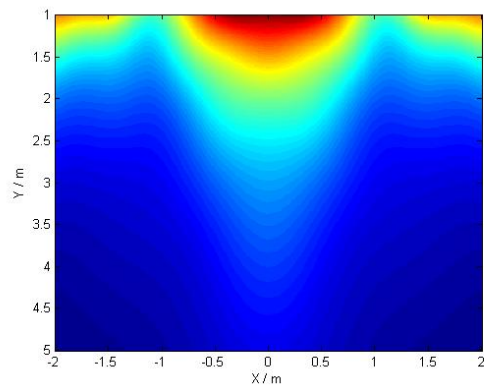
4. Simulation des Schallfeldes

der äußeren unteren Lautsprecher entsteht im Vergleich zu Schallfeld (a) jedoch keine leicht erhöhte Intensität, sondern diese steigt ausgehend von den Positionen der Lautsprecher an. Schallfeld (c) zeigt die Verteilung der Intensitäten bei einer Hörposition von 0,5 m Höhe. Wird die Verteilung der Intensität um die Position der Schallquelle betrachtet, ist ein Unterschied zu den vorherigen Schallfeldern zu erkennen. Im Gegensatz zu Schallfeld (a) und (b) steigt die Intensität zur Schallquelle hin nicht gleichmäßig an, sondern bildet zwei Maxima auf der Höhe der inneren unteren Lautsprecher. Betrachtet man die Verteilung zu den Rändern hin, entstehen auf der Höhe der äußeren unteren Lautsprecher ebenfalls Maxima.

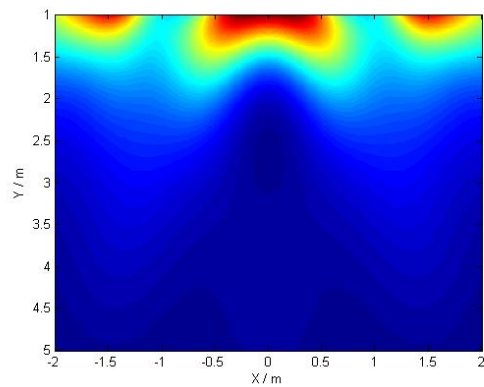
4. Simulation des Schallfeldes



(a) Hörposition auf 1,7 m



(b) Hörposition auf 1,1 m



(c) Hörposition auf 0,5 m

Abbildung 4.9.: Positionsabhängigkeit des Hörers. Schallintensität einer Schallquelle mit Koordinaten $[0, -0.1, 1.1]$, berechnet nach Gleichung 3.7; Lautsprecher auf Höhe von 2 m und mit Abstand von 0,1 m sowie Zusatzlautsprecher auf Höhe von 0,2 m und Abstand von 0,8 m; Höhenveränderung der Hörposition; Frequenz 500 Hz

Vergleicht man die simulierten Schallfelder, ist eine deutliche Veränderung zwischen den einzelnen Grafiken zu erkennen. Diese Veränderungen lassen darauf schließen, dass die Wahrnehmung einer Schallquelle von der Position des Hörers abhängt. Befindet er sich etwa in stehender Position, deutet das simulierte Schallfeld eine geringe Teilnahme der unteren Lautsprecherebene an, was wiederum eine stärkere Teilnahme der oberen Lautsprecherebene zur Folge hat und dementsprechend die Wahrnehmung der Schallquelle über ihre eigentliche Position verschiebt. Wechselt der Hörer aber in eine niedrigere Position und setzt sich beispielsweise hin, deutet das simulierte Schallfeld eine ausgeglichenerere Beteiligung beider Lautsprecherebenen an, was sich wiederum auf die Wahrnehmung der Schallquelle an ihrer eigentlichen Position auswirkt. Wird die Position des Zuhörers jedoch noch niedriger gewählt, nehmen die Lautsprecher der unteren Lautsprecherebene mehr Einfluss auf das entstehende Schallfeld als die Lautsprecher der oberen Lautsprecherebene. Daraus folgt, dass die Wahrnehmung der Schallquelle unter ihre eigentliche Position verschoben wird.

4.3. Diskussion der Resultate

Wie sich gezeigt hat, gibt es deutliche Unterschiede der simulierten Schallfelder in Abhängigkeit von der Position der Schallquelle und des Zuhörers. Befindet sich eine Schallquelle auf der Höhe der oberen Lautsprecherebene, wird ein Schallfeld berechnet, das einer sich ausbreitenden Wellenfront ähnelt. Wird die Position der Schallquelle aber in der Höhe verändert, gibt es deutliche Unterschiede der berechneten Schallfelder. Liegt die Schallquelle etwa in der Mitte der Lautsprecherebenen oder auf der Höhe der unteren Lautsprecherebene, wird keine sich gleichmäßig ausbreitende Intensität erreicht, sondern erfolgt eine ungleichmäßige Verteilung. Zurückführen lässt sich dies auf das räumliche Aliasing. In der Theorie der Wellenfeldsynthese wird von einer unendlichen Anzahl an Elementarwellen ausgegangen. Da es in der praktischen Umsetzung jedoch nicht möglich ist, eine unendliche Anzahl an Lautsprechern auf einer Ebene zu verteilen, weisen die Lautsprecher endliche Abstände zueinander auf. Diese Abstände verursachen oberhalb einer Grenzfrequenz, der Aliasfrequenz, räumliche Abbildungsfehler, woraus folgt, dass eine Schallquelle nicht mehr korrekt synthetisiert werden kann [The04]. Innerhalb des Hörfeldes der Wellenfeldsynthese-Anlage überlagern sich die durch die Lautsprecher erzeugten Elementarwellen, löschen sich gegenseitig aus oder verstärken sich. Bestimmt wird dies durch die Frequenz des Signals, den Abstand der Lautsprecher zueinander und den jeweiligen Winkel der von der Schallquelle erzeugten Wellenfront in Bezug auf die Lautsprecher. Durch diesen Effekt der Überlagerung werden die Schallfelder in Abhängigkeit von Position und Signal der Schallquelle erzeugt.

Werden die Abbildungen 4.2 und 4.3 verglichen, ergeben sich Unterschiede der simulierten Schallfelder. Das Schallfeld wird hier ausschließlich durch die obere Lautsprecherenebene synthetisiert. Bei einem Abstand der Schallquelle von 0,1 m hinter den Lautsprechern ergibt sich eine gleichmäßige Verteilung der Intensität, wohingegen bei einem Abstand von 1 m Unebenheiten zu erkennen sind. Bei einem Abstand von 1 m hinter den Lautsprechern ist jedoch ebenfalls eine gleichmäßige Verteilung zu erwarten. Das dies nicht der Fall ist, lässt sich durch die endliche Länge der Lautsprecherarrays erklären. Durch die praktisch nur begrenzt umsetzbare Theorie eines unendlichen kontinuierlichen Lautsprecherarrays, kommt es zu den dargestellten Unebenheiten des Schallfeldes.

Abbildung 4.10 stellt das räumliche Aliasing anhand einer Schallquelle mit den Koordinaten $[0, -0.1, 2]$ dar. Schallfeld (a) zeigt eine Schallquelle mit einem Signal von 500 Hz und eine sich gleichmäßig ausbreitende Intensität, während diese in Schallfeld (b) mit einem Signal von 1750 Hz deutlich zerklüftet erscheint. Ursache dafür ist der Abstand der Lautsprecher zueinander und die daraus entstehende Aliasfrequenz. Da die Schallquelle auf 2 m Höhe und somit hinter der oberen Lautsprecherenebene platziert ist, wird der größte Teil des Schallfeldes durch die obere Lautsprecherenebene synthetisiert. Der Abstand der Lautsprecher zueinander beträgt 0,1 m. Die Aliasfrequenz lässt sich näherungsweise mit der Formel

$$f_{alias} = \frac{c}{2\Delta x} \quad (4.8)$$

berechnen, wobei c die Schallgeschwindigkeit und Δx der Abstand der Lautsprecher zueinander ist [BdVV93]. Dies ergibt mit einem Lautsprecherabstand von 0,1 m eine Aliasfrequenz von etwa 1700 Hz. Wird diese Frequenz überschritten, treten Aliasing-Effekte auf, wie sie bei einer Frequenz von 1750 Hz gut erkennbar sind.

Abbildung 4.11 hingegen zeigt eine Schallquelle mit den Koordinaten $[0, -0.1, 1.1]$. Die Schallquelle befindet sich somit mittig zwischen der oberen und der unteren Lautsprecherenebene. Da sich zwischen diesen beiden Lautsprecherenebenen keine Wiedergabekanäle befinden, ist der Abstand dementsprechend groß, was sich wiederum auf die Aliasfrequenz auswirkt. Mit zunehmendem Abstand der Lautsprecher zueinander verringert sich die Aliasfrequenz. Bei einem Abstand von 0,1 m lag diese bei etwa 1700 Hz. Bei jetzigem Abstand der Lautsprecher von 1,8 m liegt die Aliasfrequenz bei etwa 100 Hz.

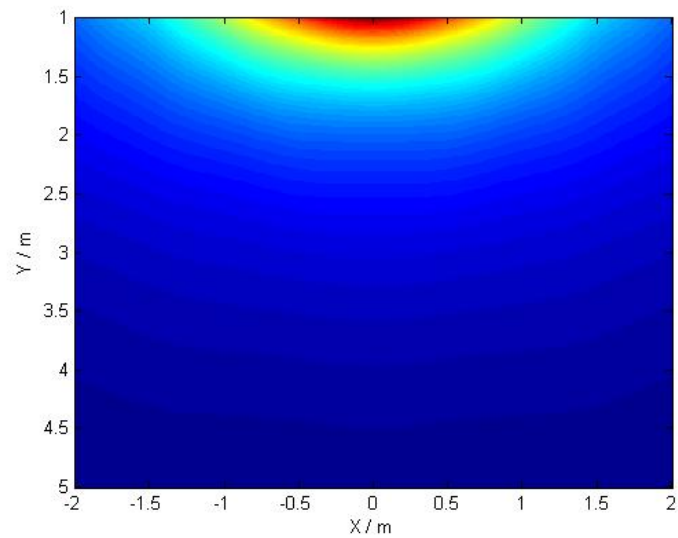
Schallfeld (a) zeigt hier eine Schallquelle mit einem Signal von 80 Hz und einer sich gleichmäßig verteilenden Intensität, während Schallfeld (b) ein Signal von 700 Hz besitzt und eine deutlich ungleichmäßige Verteilung der Intensität zeigt. Ursache hierfür ist ebenfalls das räumliche Aliasing. Wird im Vergleich die Abbildung 4.10 betrachtet, fällt auf, dass die Verteilung der Intensität unterhalb der Aliasfrequenzen gleichmäßig verläuft, oberhalb derselben aber deutli-

4. Simulation des Schallfeldes

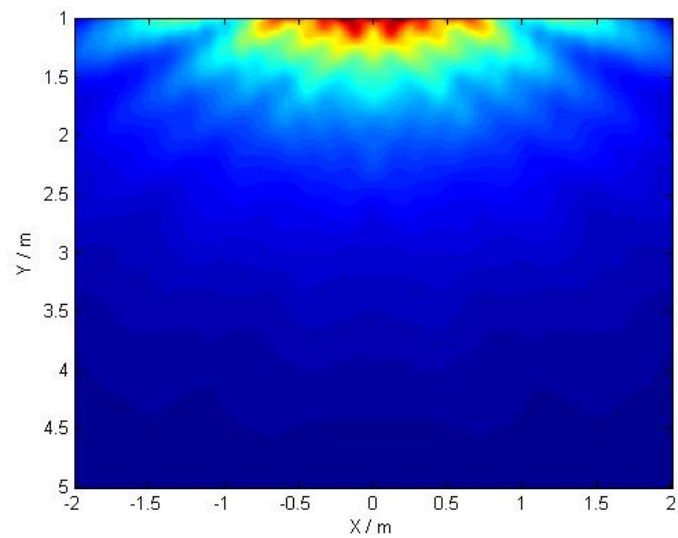
che Verformungen aufweist. Werden die jeweiligen Frequenzen miteinander verglichen, lässt sich der Einfluss der unterschiedlichen Abstände der Lautsprecher deutlich erkennen.

Wird die Position eines Zuhörers, wie in Abbildung 4.9 dargestellt, in Bezug auf eine zwischen den Lautsprecherebenen gelegene Schallquelle betrachtet, sind die Einflüsse des räumlichen Aliasings klar zu erkennen. Letztendlich zeigt dies, dass die Positionierung einer Schallquelle mit unterschiedlicher Höhenlage zwar möglich ist, jedoch bei theoretischer Betrachtung mit dem vorhandenen System keine korrekte Wellenfeldsynthese gegeben ist.

4. Simulation des Schallfeldes

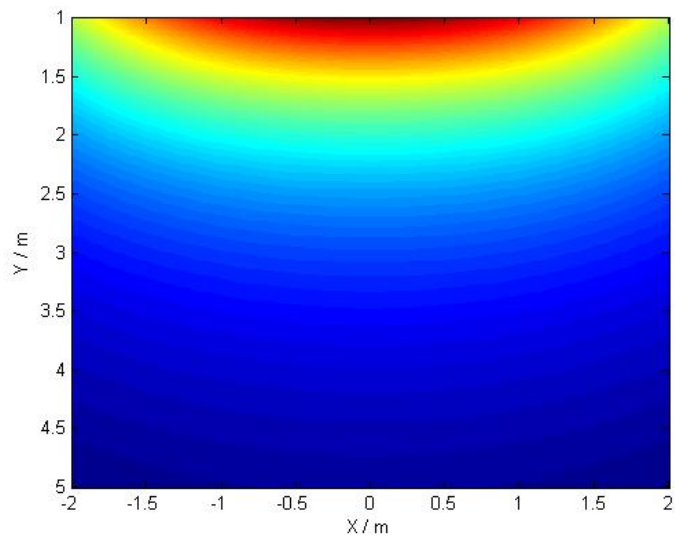


(a) Signal mit 500 Hz

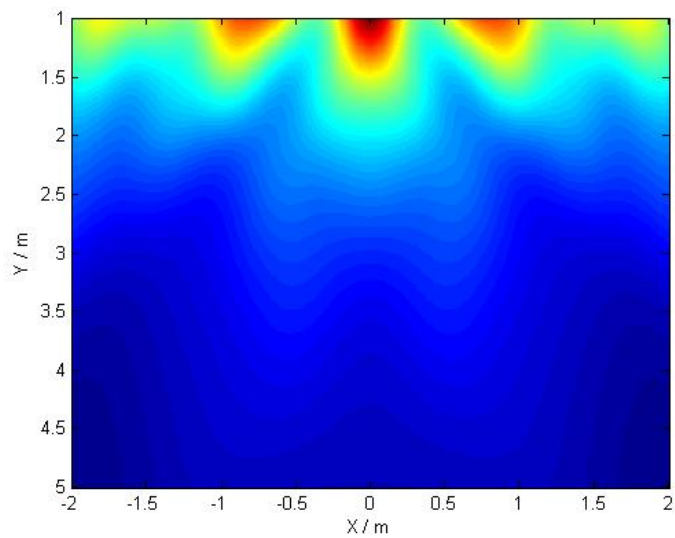


(b) Signal mit 1750 Hz

Abbildung 4.10.: Räumliches Aliasing. Schallintensität einer Schallquelle mit Koordinaten $[0, -0.1, 2]$, berechnet nach Gleichung 3.7 und mit wechselnder Frequenz; Lautsprecher auf Höhe von 2 m und mit Abstand von 0,1 m sowie Zusatzlautsprecher auf Höhe von 0,2 m und mit Abstand von 0,8 m; Aliasfrequenz 1700 Hz



(a) Signal mit 80 Hz



(b) Signal mit 700 Hz

Abbildung 4.11.: Räumliches Aliasing. Schallintensität einer Schallquelle mit Koordinaten $[0, -0.1, 1.1]$, berechnet nach Gleichung 3.7 und mit wechselnder Frequenz; Lautsprecher auf Höhe von 2 m und mit Abstand von 0,1 m sowie Zusatzlautsprecher auf Höhe von 0,2 m und mit Abstand von 0,8 m; Aliasfrequenz 100 Hz

5. Ergebnisse

Dieses Kapitel beschreibt die Messung eines von der Wellenfeldsynthese-Anlage erzeugten Schallfeldes und vergleicht dieses mit einem simulierten Schallfeld. Des Weiteren wird die Durchführung eines Hörtests für 3D-Quellenpositionen beschrieben und erfolgt eine Auswertung der Ergebnisse.

5.1. Vergleich der Simulation mit den Messdaten

Für die Durchführung der Messung wurde die Wellenfeldsynthese-Anlage der HAW Hamburg genutzt. Ziel der Messung ist der Vergleich zwischen dem real erzeugten Schallfeld einer in der dreidimensionalen Ebene positionierten Schallquelle und dem unter gleicher Voraussetzung simulierten Schallfeld. Die Messung bezieht sich dabei auf die kurze Seite der Wellenfeldsynthese-Anlage, an der sich die zusätzlichen Lautsprecher befinden.

Für die Darstellung des gemessenen Schallfeldes werden analog zur Simulation die Intensitäten an den Messpunkten benötigt. Um die Intensitäten der aufgenommenen Signale zu erhalten, wird die Raumimpulsantwort berechnet. Dies geschieht, indem weißes Rauschen über die Wellenfeldsynthese-Anlage abgespielt und anschließend über ein Mikrofon aufgenommen wird. Aus diesen Signalen wird anschließend die Kreuzkorrelation berechnet, die in diesem Fall die Raumimpulsantwort darstellt. Bei der Berechnung der Kreuzkorrelation für weißes Rauschen ist der Erwartungswert ein Signal, das symmetrisch um $lag = 0$ liegt. Um anschließend die resultierenden Intensitäten zu erhalten, werden daher nur die positiven Lags der Kreuzkorrelation aufsummiert und quadriert. Eine ausführliche Betrachtung findet sich in Görnes *Tontechnik* [Gör08].

5.1.1. Durchführung der Messung

Um das von der Wellenfeldsynthese-Anlage erzeugte Schallfeld auswerten zu können, werden innerhalb des von der Anlage erzeugten Hörbereichs in einem Bereich von 4×3 m an bestimmten Messpunkten Messungen durchgeführt. Der Messbereich wird dabei durch die Koordinaten $[-2, 1]$, $[2, 1]$, $[-2, 4]$ und $[2, 4]$ aufgespannt. Die Messpunkte befinden sich in

x-Richtung in einem Abstand von 0,1 m, in y-Richtung in einem Abstand von 1 m sowie in einer Höhe von 1,7 m. Insgesamt ergibt sich so ein gemessenes Schallfeld aus 164 einzelnen Messpositionen. Die Positionen der Messpunkte werden anhand des Trackingsystems erfasst. Für die Durchführung der Messung wurde ein Studiomikrofon vom Typ AKG C 3000 B genutzt. Um mit diesem Mikrofon die von der Wellenfeldsynthese-Anlage erzeugten Signale aufnehmen zu können, muss es mit dem System verbunden werden. Hierfür wird das RedNet-Ethernet-Netzwerk-Studio-Interface genutzt, über das auch die zusätzlichen Lautsprecherkomponenten mit dem System verbunden sind. Dieses Interface bietet neben acht analogen Ausgängen ebenfalls acht analoge Eingänge, über die das Studiomikrofon mit dem System verbunden wird. Zu beachten ist hier, dass das Interface über keine eigene Phantomspeisung verfügt und daher ein zusätzlicher Mikrofon-Vorverstärker genutzt werden muss.

Für die Messung wird eine virtuelle Schallquelle an den Koordinaten $[0, -1, 1.1]$ positioniert. Die Schallquelle befindet sich somit einen Meter hinter und exakt mittig zwischen den Lautsprecherebenen. Mittels der Digital Audio Workstation Cubase wird das von Audacity generierte, 10 Sekunden lange weiße Rauschen durch diese Schallquelle wiedergegeben und gleichzeitig über das Mikrofon aufgenommen.

5.1.2. Auswertung der Messdaten

Mit der Software MATLAB wurde ein Verfahren zur Auswertung der Messdaten entwickelt. Um die als Audiodateien vorliegenden Messdaten auswerten zu können, werden diese von MATLAB eingelesen und stehen anschließend innerhalb der Software als Datenarray zur Verfügung. Hier wird für jede Audiodatei, die jeweils einen Messpunkt darstellt, die Kreuzkorrelation mit dem Eingangssignal, dem weißen Rauschen, berechnet. Um die entsprechenden Intensitäten an den jeweiligen Messpunkten zu erhalten, werden die positiven Lags der Kreuzkorrelation bestimmt, aufsummiert und quadriert. Als Ergebnis entsteht ein Plot, der die entstandenen Intensitäten an den einzelnen Messpunkten darstellt. Abbildung 5.1 stellt den Ablauf des Verfahrens dar. Abbildung 5.2 zeigt das resultierende Messergebnis. Wird die Verteilung der Intensitäten bei y-Position 1 betrachtet, sind Intensitätsmaxima auf der Höhe der Position der virtuellen Schallquelle zu erkennen. Im weiteren Verlauf fällt auf, dass die Intensität zu den Rändern hin abfällt, jedoch auf der Höhe der x-Positionen -1.5 und 1.5 wieder kurzzeitig ansteigt. Eine ähnliche Verteilung ergibt sich bei y-Position 2. Hier steigt die Intensität im Verlauf zur Position der Schallquelle hin an, fällt jedoch auf der Höhe der Schallquelle etwas ab. Je größer der Abstand zur Schallquelle wird, desto ebener wird die Verteilung der Intensitäten. Dies macht sich bei den y-Positionen 3 und 4 bemerkbar. Hier sind zwar geringe Anstiege der

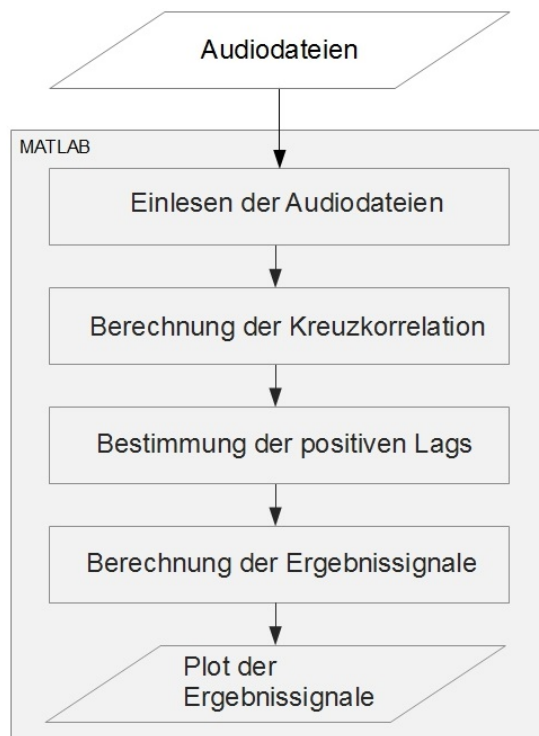


Abbildung 5.1.: Ablauf zur Auswertung der Messdaten

Intensität zur Position der Schallquelle hin zu erkennen, jedoch fallen diese nicht so stark aus, wie es bei den y-Positionen 1 und 2 der Fall ist.

Abbildung 5.3 zeigt das simulierte Schallfeld. Die virtuelle Schallquelle befindet sich dabei an derselben Position wie bei der Messung. Wird das simulierte Schallfeld betrachtet, fällt ein deutliches Maximum der Intensität auf der Position der Schallquelle auf, das im Verlauf zu den Rändern hin stark abfällt. Wird der weitere Verlauf genauer betrachtet, fällt die Intensität erst ab, steigt jedoch anschließend zwischen den x-Positionen -1 und -1.5 , sowie 1 und 1.5 wieder leicht an.

Ein Vergleich der beiden entstandenen Schallfelder gestaltet sich schwierig, da das simulierte Schallfeld im Gegensatz zum gemessenen deutlich präziser dargestellt werden kann. Vergleicht man sie dennoch, sind Gemeinsamkeiten erkennbar. So bilden sich bei beiden Schallfeldern Maxima der Intensität auf der Höhe der Position der virtuellen Schallquelle. Wird der Verlauf ausgehend von der Schallquelle zu den Rändern hin betrachtet, fällt bei beiden Darstellungen die Intensität erst ab, steigt an den x-Positionen -1 und 1 aber wieder leicht an. Zu erklären sind diese leichten Anstiege der Intensität durch die am System teilnehmenden Zusatzlautsprecher, die die untere Lautsprecherebene bilden.

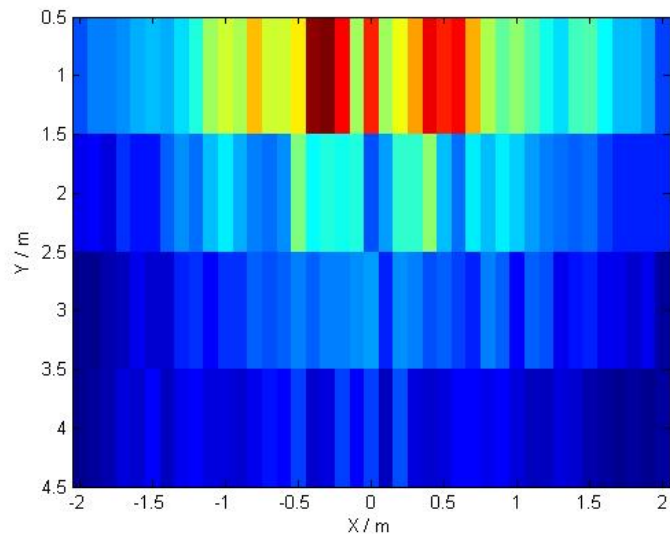


Abbildung 5.2.: Gemessene Schallintensität einer Schallquelle mit Koordinaten $[0, -1, 1.1]$; Lautsprecher auf Höhe von 2 m und mit Abstand von 0,1 m sowie Zusatzlautsprecher auf Höhe von 0,2 m und mit Abstand von 0,8 m

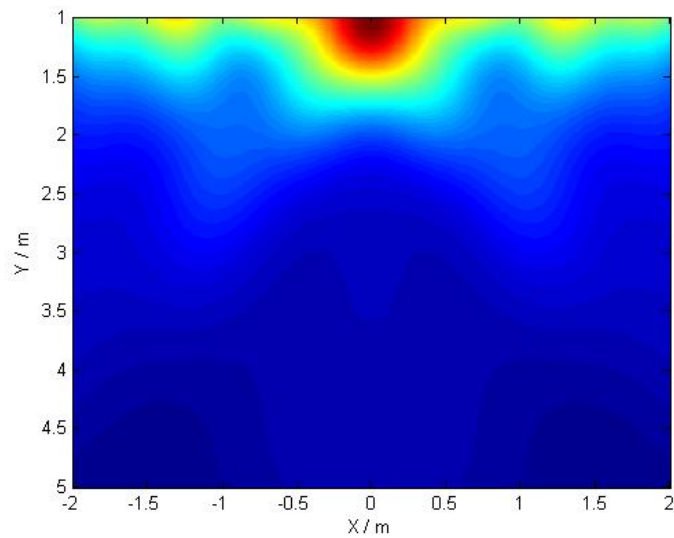


Abbildung 5.3.: Simulierte Schallintensität einer Schallquelle mit Koordinaten $[0, -1, 1.1]$, berechnet nach Gleichung 3.7; Lautsprecher auf Höhe von 2 m und mit Abstand von 0,1 m sowie Zusatzlautsprecher auf Höhe von 0,2 m und mit Abstand von 0,8 m; Frequenz 500 Hz

Betrachtet man die x-Positionen, befinden sich diese Anstiege der Intensität auf der Höhe der Positionen der äußeren Zusatzlautsprecher, die sich an den x-Positionen -1.2230 und 1.2230 befinden.

5.2. Hörtest für 3D-Quellenpositionen

Mit der erweiterten Version der WONDER-Software ist es möglich, eine virtuelle Schallquelle in dreidimensionaler Richtung zu positionieren. Die Simulation unterschiedlicher Schallfelder hat bereits gezeigt, dass eine dreidimensionale Positionierung einer virtuellen Schallquelle möglich ist, die dadurch entstandenen Schallfelder jedoch nicht auf eine korrekte Synthese einer Wellenfront schließen lassen. Da die Simulation allerdings nur eine theoretische Betrachtung darstellt, sollen mittels eines Hörtests Probanden auf die Wahrnehmung einer so positionierten Schallquelle getestet werden. Ziel dieses Tests ist es, den real entstehenden Höreindruck einer solchen Schallquelle zu ermitteln. Der Test besteht dabei aus verschiedenen positionierten Schallquellen, die sich in unterschiedlichen Richtungen zwischen den Lautsprecherebenen bewegen. Die Probanden bekommen den Auftrag, den räumlichen Höreindruck und eventuelle Auffälligkeiten für zwölf Testfälle zu beschreiben. Der zugehörige Fragebogen befindet sich im Anhang.

5.2.1. Versuchsdurchführung

Um den bestmöglichen Höreindruck erreichen zu können, wird die Testperson gebeten, sich mittig innerhalb des von der Wellenfeldsynthese-Anlage aufgespannten Hörfeldes auf einen Stuhl zu setzen. Das Gehör befindet sich so auf etwa $1,1$ m Höhe, was der Mitte der beiden Lautsprecherebenen entspricht. Auf dieser Höhe treffen die von den Lautsprechern erzeugten Signale exakt aufeinander und erzeugen das bestmögliche Signal. Anschließend werden verschiedene Testfälle mit unterschiedlicher Quellenpositionierung durchgespielt. Bei den Testfällen handelt es sich jeweils um eine Schallquelle, die sich von einer bestimmten Position zu einer anderen bewegt. Tabelle 5.1 gibt einen Überblick über das Testszenario.

Für die Durchführung der unterschiedlichen Testfälle werden die Schallquellen über von Super-Collider versendete OSC-Nachrichten gesteuert. Da sich die Erweiterung der Wellenfeldsynthese-Anlage auf vier Zusatzlautsprecher beschränkt, orientieren sich die Positionen der Schallquellen an den äußeren Lautsprechern der unteren Lautsprecherebene. Somit werden die Schallquellen nicht am äußeren Rand des von der Wellenfeldsynthese aufgespannten Hörfeldes platziert, sondern jeweils auf der Höhe der äußeren unteren Lautsprecher.

5. Ergebnisse

Testfall	Weg der Schallquelle	Verfahren	Sound
1	links oben nach rechts unten	Stereo-Panning	Hubschrauber
2	mittig rechts nach links	Stereo-Panning	Hubschrauber
3	rechts unten nach links oben	Stereo-Panning	Hubschrauber
4	rechts oben nach links unten	Stereo-Panning	Spatz
5	mittig links nach rechts	Stereo-Panning	Spatz
6	links unten nach rechts oben	Stereo-Panning	Spatz
7	links oben nach rechts unten	3D- Quellenpositionierung	Hubschrauber
8	mittig rechts nach links	3D- Quellenpositionierung	Hubschrauber
9	rechts unten nach links oben	3D- Quellenpositionierung	Hubschrauber
10	rechts oben nach links unten	3D- Quellenpositionierung	Spatz
11	mittig links nach rechts	3D- Quellenpositionierung	Spatz
12	rechts unten nach links oben	3D- Quellenpositionierung	Spatz

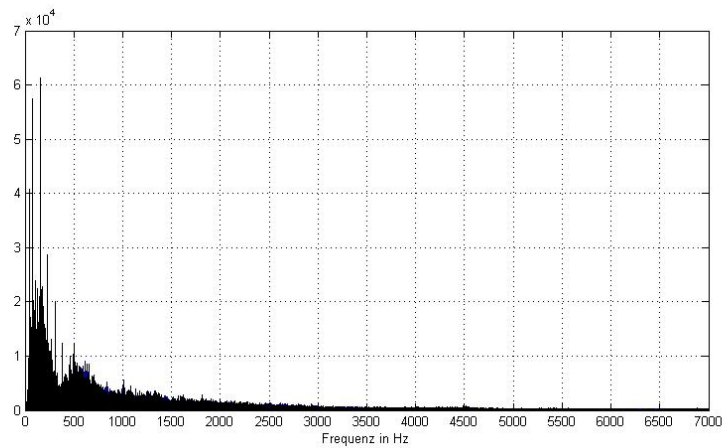
Tabelle 5.1.: Gesamtes Testszenario mit Weg der Schallquelle, Verfahren der Audiowiedergabe und des Sounds

Da die unteren Lautsprecher jeweils mittig unter einem oberen Lautsprechermodul platziert wurden und dementsprechend unter den äußeren Lautsprechermodulen keine Zusatzlautsprecher vorhanden sind, soll dieses Vorgehen dazu dienen, einen ausgeglichenen Höreindruck zu schaffen.

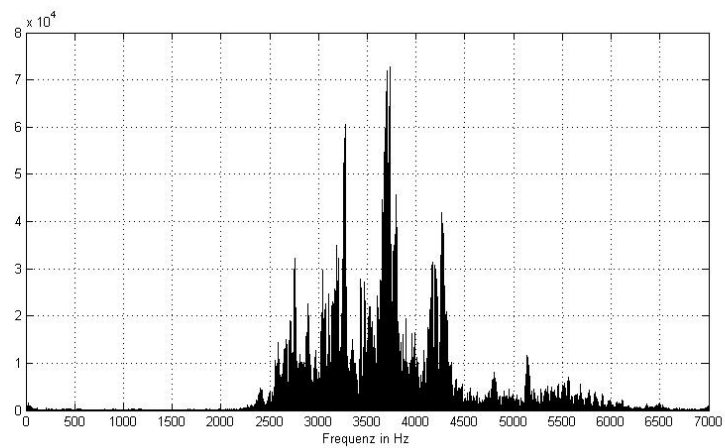
Um eine Schallquelle im Raum bewegen zu können, werden bei der Durchführung zwei unterschiedliche Verfahren genutzt. Einerseits ist dies das Stereo-Panning, bei dem zwei Schallquellen einander gegenüberliegend positioniert werden und die Lautstärke der Audiospuren so angepasst wird, dass sie bei der einen Schallquelle konstant abnimmt und bei der anderen konstant steigt. Die dadurch entstehende Phantomschallquelle bewegt sich durch die Pegel-differenz der Audiospuren von der Position der einen Schallquelle zur Position der anderen Schallquelle. Andererseits wird die 3D-Quellenpositionierung der WONDER-Software ein-

gesetzt. Hierzu wird eine Schallquelle an einer bestimmten Stelle positioniert. Mittels der WONDER-Software ist es möglich, diese Schallquelle über einen bestimmten Zeitraum von einer Position zu einer anderen laufen zu lassen. Dies wird genutzt, um die Schallquelle im Raum zu bewegen. Die erste Hälfte der Testfälle wird mittels Stereo-Panning realisiert, die zweite Hälfte mittels der 3D-Quellenpositionierung. Ziel der unterschiedlichen Verfahren soll sein, eventuelle Auffälligkeiten und Unterschiede bei der Verwendung dieser Verfahren zu erkennen.

Da das von der Wellenfeldsynthese-Anlage erzeugte Schallfeld innerhalb des Bereiches zwischen den Lautsprechererebenen auf Grund des räumlichen Aliasings stark von der jeweiligen Frequenz des Signals abhängt, werden zwei unterschiedliche Signale für die Testfälle genutzt. Als tieffrequentes Signal dient für die Testfälle 1 bis 3 sowie 7 bis 9 der Sound eines Hubschraubers, als hochfrequentes Signal für die verbleibenden Testfälle der Ruf eines Spatzes. Durch den Abstand der Lautsprechererebenen zueinander wird eine niedrige Aliasfrequenz erreicht, die sich wiederum auf das entstehende Schallfeld auswirkt. Anhand der vom Frequenzspektrum unterschiedlichen Signale soll überprüft werden, wie sich das räumliche Aliasing tatsächlich auf den entstehenden Höreindruck auswirkt. Theoretisch betrachtet ist das Signal des Hubschraubers aufgrund der hauptsächlich tiefen und somit näher an der Aliasfrequenz liegenden Frequenzanteile besser lokalisierbar als das Signal des Spatzes. Abbildung 5.4 zeigt die unterschiedlichen Frequenzspektren. Das Signal des Hubschraubers setzt sich hauptsächlich aus Frequenzen unterhalb von 500 Hz zusammen, das Signal des Spatzes hingegen aus Frequenzen um die 3500 Hz.



(a) Hubschrauber



(b) Spatz

Abbildung 5.4.: Frequenzspektren der Signale von Hubschrauber und Spatz. Frequenzen bei Hubschrauber hauptsächlich unter 500 Hz, bei Spatz um 3500 Hz

5.2.2. Auswertung der Ergebnisse

Das Testszenario wurde mit insgesamt acht Probanden durchgeführt. Für die Auswertung der Ergebnisse wurde der Anteil der richtig beschriebenen Richtungsverläufe der Schallquelle prozentual ausgerechnet. Dabei wurde nur die von der Testperson korrekt erkannte Richtung der sich bewegenden Schallquelle mit einbezogen. Im Anschluss daran wurden die von den Testpersonen beschriebenen Auffälligkeiten in Bezug auf die Richtungsänderung der Schallquelle ausgewertet. Da diese Auffälligkeiten bei jeder Testperson unterschiedlich ausfielen, erschien es sinnvoll, sich auf Gemeinsamkeiten der Aussagen zu beschränken. Tabelle 5.2 zeigt die Auswertung der Testergebnisse. Dargestellt ist die Rate der Testpersonen in Prozent, die die Richtung der sich bewegenden Schallquelle korrekt erkannt haben, sowie eine Zusammenfassung der Aussagen in Bezug auf die beschriebenen Auffälligkeiten.

Testfall	Richtig erkannt in %	Beschriebene Auffälligkeiten
1 li. o -> re. u / h	50	Schwankung der Lautstärke
2 m re. -> li. / h	63	
3 re. u -> li. o / h	50	Schwankung der Lautstärke
4 re. o -> li. u / s	38	Lokalisation schwer zu deuten
5 m li. -> re. / s	50	Lokalisation schwer zu deuten
6 li. u -> re. o / s	63	Lokalisation schwer zu deuten
7 li. o -> re. u / h	50	Schwankung der Lautstärke
8 m re. -> li. / h	38	
9 re. u -> li. o / h	50	Schwankung der Lautstärke
10 re. o -> li. u / s	38	Lokalisation schwer zu deuten
11 m li. -> re. / s	63	Lokalisation schwer zu deuten
12 re. u -> li. o / s	50	Lokalisation schwer zu deuten

Tabelle 5.2.: Auswertung des Testszenarios mit Angabe der richtig erkannten Bewegungsrichtung der Schallquelle sowie Zusammenfassung der beschriebenen Auffälligkeiten; li. links, re. rechts, m mittig, u unten, o oben, h Helikopter, s Spatz

Betrachtet man die Rate der richtig erkannten Bewegungsrichtungen der Schallquelle und bildet den Mittelwert in Bezug auf das Stereo-Panning, die 3D-Quellenpositionierung und

alle Testfälle, liegt dieser beim Stereo-Panning bei einem Wert von etwa 52 %, bei der 3D-Quellenpositionierung von etwa 48 % und bei Betrachtung aller Testfälle bei etwa 50 %.

Werden die von den Testpersonen beschriebenen Auffälligkeiten betrachtet, gibt es auffallende Gemeinsamkeiten. So wurden bei den diagonal verlaufenden Schallquellen mit dem Sound eines Hubschraubers sowohl bei dem Stereo-Panning-Verfahren als auch bei der 3D-Quellenpositionierung Schwankungen der Lautstärke festgestellt. Bei der mittig gelegenen horizontalen Bewegung einer Schallquelle, die den Sound eines Hubschraubers wiedergibt, wurden keinerlei Schwankungen der Lautstärke oder sonstige Auffälligkeiten angemerkt. Bei den Testfällen mit dem Sound eines Spatzes hingegen wurden bei allen Bewegungsrichtungen Schwierigkeiten bei der Lokalisation der Schallquelle angemerkt.

Werden die Prozentwerte der richtig erkannten Bewegungsrichtungen mit den beschriebenen Auffälligkeiten verglichen, lassen sich keine eindeutigen Zusammenhänge feststellen. Betrachtet man die Testfälle 4, 5 und 6, die den Sound des Spatzes beinhalteten, so wurde bei allen eine schwer zu deutenden Lokalisation angemerkt, jedoch liegen die Erkennungsraten von 5 und 6 höher als die von 4. Dasselbe Verhalten lässt sich bei den Testfällen 10, 11 und 12 feststellen. Auffallend ist hier die höhere Erkennungsrate im Vergleich der Richtungsbewegung von den Testfällen 4 und 6 sowie 10 und 12. Diese liegt bei einer Bewegung von unten nach oben höher als bei einer Bewegung von oben nach unten.

Werden hingegen die Testfälle 1, 2 und 3 betrachtet, die den Sound des Hubschraubers beinhalteten, ist die Erkennungsrate bei 1 und 3 gleich, jedoch bei 2 höher. Dasselbe lässt sich wiederum bei den Testfällen 7, 8 und 9 feststellen, allerdings mit dem Unterschied, dass hier bei Testfall 8 die Erkennungsrate niedriger liegt. Bemerkenswert sind hier die beschriebenen Auffälligkeiten der schwankenden Lautstärke bei den Testfällen 1, 3, 7 und 9 mit jeweils derselben Erkennungsrate, während die Testfälle 2 und 8 ohne beschriebene Auffälligkeiten eine deutlich unterschiedliche Erkennungsrate aufweisen.

Die Mehrzahl an Testpersonen beschrieb nach Abschluss des Testszenarios eine unterschiedliche Wahrnehmung des Schalls in Abhängigkeit von der Position des Gehörs. So wurde angemerkt, dass sich die Wahrnehmung bei Drehen oder Senken des Kopfes verändere.

Eine eindeutige Schlussfolgerung lässt sich aus dem Hörtest nicht ziehen. Ziel war es, den real entstehenden Höreindruck einer sich in verschiedene Richtungen bewegendem Schallquelle zu ermitteln. Die Erkennungsrate der korrekt erkannten Bewegungsrichtungen liegt bei etwa 50 %. Dies deutet zwar eine teils korrekte Wahrnehmung der Testpersonen an, jedoch war sie zu einem gleich großen Teil nicht korrekt. Deutlich hingegen lässt sich der Einfluss des räumlichen Aliasings erkennen. Bei den Testfällen mit dem hochfrequenten Signal des Spatzes treten Schwierigkeiten in der Lokalisation auf. Dies deutet auf eine nicht korrekte Synthese der

Wellenfront hin, bedingt durch die niedrige Aliasfrequenz, die durch die Abstände der Lautsprecher zueinander entsteht. Ebenso deutet die von den Testpersonen angemerkte unterschiedliche Wahrnehmung des Schalls in Abhängigkeit von der Position des Kopfes darauf hin. Bestätigt wird dies durch die simulierte Positionsabhängigkeit des Hörers in Kapitel 4, Abbildung 4.9. Dort ist ein eindeutig wechselndes Schallfeld in Abhängigkeit von der Position des Hörers zu erkennen. Beim tieffrequenten Signal des Hubschraubers lassen sich die Schwankungen der Lautstärke durch die geringe Anzahl der Lautsprecher auf der unteren Lautsprecherebene und den Abstand zwischen den Lautsprecherebenen erklären. Die Anzahl an Lautsprechern der unteren Ebene ist zu gering, um einen gleichmäßigen Höreindruck zu schaffen. Ein Unterschied in der Wahrnehmung in Bezug auf die verwendeten Wiedergabeverfahren lässt sich nicht feststellen.

Letztendlich lässt sich sagen, dass mit lediglich vier Zusatzlautsprechern eine korrekte Wellenfeldsynthese, wie schon in Kapitel 4 dargestellt, nicht möglich ist. Es lässt sich zwar ansatzweise die Wahrnehmung eines dreidimensionalen Höreindrucks durch den Hörtest bestätigen, jedoch ist die Wahrnehmung bedingt durch das räumliche Aliasing und die geringe Anzahl an zusätzlichen Lautsprechern deutlich gestört.

6. Fazit

6.1. Zusammenfassung

Ziel dieser Arbeit war es, die bestehende Version der WONDER-Software um eine 3D-Quellenposition und zusätzliche Lautsprecherkomponenten zu erweitern. Zu Beginn wurde in Kapitel 2 ein kurzer Überblick über das Prinzip der Wellenfeldsynthese gegeben, um danach das Labor der HAW Hamburg, in dem eine Wellenfeldsynthese-Anlage installiert ist, zu beschreiben und die technische Erweiterung um die zusätzlichen Lautsprecherkomponenten zu erläutern. Im Anschluss wurde näher auf die in diesem System verwendete Software WONDER eingegangen. Die für eine Erweiterung um 3D-Quellenpositionen nötigen Softwarekomponenten wurden beschrieben und die Erweiterungen implementiert.

Um einen Eindruck von den durch diese Erweiterungen entstehenden Schallfeldern zu bekommen, wurde ein Verfahren entwickelt, das die von den teilnehmenden Lautsprechern ausgesendeten Audiosignale berechnet und die daraus entstehenden Intensitäten in einer Grafik darstellt. Anschließend wurden für verschiedene Quellenpositionen Grafiken erzeugt und die Resultate diskutiert.

Zuletzt wurde eine Messung des real entstehenden Schallfeldes vorgenommen und mit der Simulation verglichen sowie ein Hörtest mit Probanden durchgeführt, um die tatsächliche Wahrnehmung einer 3D-Quellenposition zu ermitteln.

6.2. Fazit

In dieser Arbeit wurde gezeigt, dass eine Erweiterung der Wellenfeldsynthese-Software WONDER um 3D-Quellenpositionen und zusätzliche Lautsprecherkomponenten möglich ist. Die von der Wellenfeldsynthese-Anlage erzeugten Schallfelder einer dreidimensional positionierten Schallquelle deuten hingegen auf eine nicht korrekte Synthese der Wellenfront hin. Deutlich wird dies einerseits durch die Simulation der Schallfelder mit unterschiedlicher Positionierung der Schallquelle. Die erzeugten Grafiken zeigen bei einer in der Höhe veränderten Schallquelle deutlich unkorrekt synthetisierte Schallfelder. Andererseits wird dies durch den Höreindruck von Probanden bestätigt, die eine zumeist gestörte Wahrnehmung der Schallquelle anmerkten.

Ursächlich hierfür ist die Erweiterung um lediglich vier zusätzliche Lautsprecherkomponenten. Mit einer derart geringen Anzahl an zusätzlichen Wiedergabekanälen und dem daraus resultierenden räumlichen Aliasing lässt sich keine korrekte Wellenfeldsynthese in dritter Dimension durchführen.

6.3. Ausblick

Die WONDER-Software berechnet für jeden am System teilnehmenden Lautsprecher ein eigenes Audiosignal. Dabei wird keine Rücksicht auf die Abstände der Lautsprecher zueinander genommen. Um den Höreindruck bei einer 3D-Quellenpositionierung zu verbessern, kann WONDER derart erweitert werden, dass die Abstände der Lautsprecher zueinander berücksichtigt und durch die Pegel- oder Laufzeitdifferenz, wie es bei der Stereophonie der Fall ist, ausgeglichen werden. Analog dazu kann die aus vier zusätzlichen Lautsprechern bestehende untere Lautsprecher Ebene um weitere Lautsprecher ergänzt werden. Der Idealfall wäre eine untere Lautsprecher Ebene nach demselben Aufbau wie die obere Lautsprecher Ebene. Somit beschränkt sich die Nutzung der Pegel- und Laufzeitdifferenz auf den Bereich zwischen den Lautsprecher Ebenen. Eine theoretische Betrachtung dieses Verfahrens findet sich in [Mon11]. Als weiteres Testmodell könnte die Unterstützung durch visuelle Medien dienen. Bekommt ein Zuhörer zu einem bestimmten Sound ein passendes Bild geliefert, könnte sich dies positiv auf die Wahrnehmung des erwünschten Höreindrucks auswirken. Des Weiteren kann untersucht werden, ob die Wahrnehmung einer Schallquelle besser wird, wenn diese direkt auf einem Lautsprecher positioniert ist, anstatt sich in der Ebene zwischen mehreren Lautsprechern zu befinden.

Außerdem kann ein neues Verfahren der Visualisierung entwickelt werden. Setzt sich die 3D-Quellenpositionierung durch, ist eine grafische Positionierung der Schallquelle für den Anwender deutlich angenehmer. Hierzu kann die bestehende Softwarekomponente *xwonder* erweitert oder eine gänzlich neue Anwendung entworfen werden.

A. Benutzertest und Messpunkte

Fragebogen

Bachelorarbeit
Marvin Geitner

	Ereignis (Welchen Weg nahm die Schallquelle?)	Auffälligkeiten (Schwankung der Lautstärke, Schärfe der Lokalisation)
Test 1 links oben nach rechts unten		
Test 2 rechts nach links		
Test 3 rechts unten nach links oben		
Test 4 rechts oben nach links unten		
Test 5 links nach rechts		
Test 6 links unten nach rechts oben		
Test 7 links oben nach rechts unten		
Test 8 rechts nach links		
Test 9 rechts unten nach links oben		
Test 10 rechts oben nach links unten		
Test 11 links nach rechts		
Test 12 rechts unten nach links oben		

A.1. SuperCollider Skript

```
1
2 //1. Test Stereo-Panning. Hubschrauber von links o. nach rechts u.
3
4 NetAddr("192.168.3.1", 58100)
5     .sendMsg("/WONDER/source/position3D", 0, -1.2230, -2, 2);
6
7 NetAddr("192.168.3.1", 58100)
8     .sendMsg("/WONDER/source/position3D", 1, 1.2230, -2, 0.2);
9
10 //2. Test Stereo-Panning. Hubschrauber von rechts nach links.
11
12 NetAddr("192.168.3.1", 58100)
13     .sendMsg("/WONDER/source/position3D", 0, 1.2230, -1.0, 2);
14
15 NetAddr("192.168.3.1", 58100)
16     .sendMsg("/WONDER/source/position3D", 1, 1.2230, -1.0, 0.2);
17 //----
18 NetAddr("192.168.3.1", 58100)
19     .sendMsg("/WONDER/source/position3D", 0, -1.2230, -1.0, 2, 38);
20
21 NetAddr("192.168.3.1", 58100)
22     .sendMsg("/WONDER/source/position3D", 1, -1.2230, -1.0, 0.2, 38);
23
24 //3. Test Stereo-Panning. Hubschrauber von rechts u. nach links o.
25
26 NetAddr("192.168.3.1", 58100)
27     .sendMsg("/WONDER/source/position3D", 0, 1.2230, -2, 0.2);
28
29 NetAddr("192.168.3.1", 58100)
30     .sendMsg("/WONDER/source/position3D", 1, -1.2230, -2, 2);
31
32 //4. Test Stereo-Panning. Spatz von rechts o. nach links u.
33
34 NetAddr("192.168.3.1", 58100)
35     .sendMsg("/WONDER/source/position3D", 0, 1.2230, -1.6, 2);
36
37 NetAddr("192.168.3.1", 58100)
38     .sendMsg("/WONDER/source/position3D", 1, -1.2230, -1.6, 0.2);
```

```
39
40 //5. Test Stereo-Panning. Spatz von links nach rechts.
41
42 NetAddr("192.168.3.1", 58100)
43     .sendMsg("/WONDER/source/position3D", 0, -1.2230, -2, 2);
44
45 NetAddr("192.168.3.1", 58100)
46     .sendMsg("/WONDER/source/position3D", 1, -1.2230, -2, 0.2);
47 //----
48 NetAddr("192.168.3.1", 58100)
49     .sendMsg("/WONDER/source/position3D", 0, 1.2230, -2, 2, 26);
50
51 NetAddr("192.168.3.1", 58100)
52     .sendMsg("/WONDER/source/position3D", 1, 1.2230, -2, 0.2, 26);
53
54 //6. Test Stereo-Panning. Spatz von links u. nach rechts o.
55
56 NetAddr("192.168.3.1", 58100)
57     .sendMsg("/WONDER/source/position3D", 0, -1.2230, -1.6, 0.2);
58
59 NetAddr("192.168.3.1", 58100)
60     .sendMsg("/WONDER/source/position3D", 1, 1.2230, -1.6, 2);
61
62 //7. Test WONDER-3D. Hubschrauber von links o. nach rechts u.
63
64 NetAddr("192.168.3.1", 58100)
65     .sendMsg("/WONDER/source/position3D", 0, -1.2230, -1.4, 2);
66
67 NetAddr("192.168.3.1", 58100)
68     .sendMsg("/WONDER/source/position3D", 0, 1.2230, -1.4, 0.2, 38);
69
70 //8. Test WONDER-3D. Hubschrauber von rechts nach links.
71
72 NetAddr("192.168.3.1", 58100)
73     .sendMsg("/WONDER/source/position3D", 0, 1.2230, -1.5, 1.1);
74
75 NetAddr("192.168.3.1", 58100)
76     .sendMsg("/WONDER/source/position3D", 0, -1.2230, -1.5, 1.1, 38);
77
78 //9. Test WONDER-3D. Hubschrauber von rechts u. nach links o.
```



```
79
80 NetAddr("192.168.3.1", 58100)
81     .sendMsg("/WONDER/source/position3D", 0, 1.2230, -1.4, 0.2);
82
83 NetAddr("192.168.3.1", 58100)
84     .sendMsg("/WONDER/source/position3D", 0, -1.2230, -1.4, 2, 38);
85
86 //10. Test WONDER-3D. Spatz von rechts o. nach links u.
87
88 NetAddr("192.168.3.1", 58100).
89     sendMsg("/WONDER/source/position3D", 0, 1.2230, -1.5, 2);
90
91 NetAddr("192.168.3.1", 58100).
92     sendMsg("/WONDER/source/position3D", 0, -1.2230, -1.5, 0.2, 22);
93
94 //11. Test WONDER-3D. Spatz von links nach rechts.
95
96 NetAddr("192.168.3.1", 58100)
97     .sendMsg("/WONDER/source/position3D", 0, -1.2230, -1.5, 1.1);
98
99 NetAddr("192.168.3.1", 58100)
100     .sendMsg("/WONDER/source/position3D", 0, 1.2230, -1.5, 1.1, 22);
101
102 //12. Test WONDER-3D. Spatz von rechts u. nach links o.
103
104 NetAddr("192.168.3.1", 58100)
105     .sendMsg("/WONDER/source/position3D", 0, 1.2230, -1.5, 0.2);
106
107 NetAddr("192.168.3.1", 58100)
108     .sendMsg("/WONDER/source/position3D", 0, -1.2230, -1.5, 2, 22);
```

Listing A.1: SuperCollider Skript zur Steuerung der Schallquellen des Hörtests

A.2. Messpunkte

A. Benutzertest und Messpunkte

Messreihe 1			
X	Y	X	Y
-1606	1356	502	1347
-1505	1357	606	1343
-1402	1356	682	1358
-1297	1354	779	1349
-1208	1352	883	1342
-1108	1357	991	1345
-1007	1352	1083	1349
-903	1352	1186	1347
-815	1347	1287	1344
-705	1344	1379	1345
-607	1350	1483	1345
-509	1346	1581	1348
-407	1348	1686	1346
-300	1349	1783	1345
-195	1349	1886	1347
-115	1351	1985	1346
-1	1354	2086	1348
98	1348	2187	1347
197	1344	2284	1346
296	1347	2380	1346
399	1354		

Messreihe 2			
X	Y	X	Y
-1675	386	471	388
-1565	389	586	383
-1470	389	683	379
-1373	380	784	386
-1274	387	892	386
-1179	383	989	383
-1076	382	1092	378
-970	384	1193	380
-876	389	1292	386
-778	390	1393	385
-672	383	1490	379
-576	384	1598	382
-471	383	1689	384
-368	381	1782	386
-263	383	1875	383
-164	385	1986	382
-68	382	2078	381
-67	383	2178	374
169	383	2280	386
269	376	2379	383
369	385		

Messreihe 3			
X	Y	X	Y
-1676	-391	496	-403
-1568	-390	594	-405
-1455	-406	696	-395
-1353	-405	797	-398
-1247	-405	891	-403
-1144	-411	986	-393
-1048	-390	1090	-392
-936	-386	1185	-394
-843	-386	1288	-393
-744	-383	1381	-393
-645	-384	1482	-390
-550	-390	1581	-397
-442	-388	1688	-391
-344	-382	1783	-385
-244	-382	1887	-382
-149	-381	1982	-397
-50	-383	2009	-396
47	-387	2187	-394
153	-384	2280	-395
197	-385	2379	-396
398	-395		

Messreihe 4			
X	Y	X	Y
-1675	-1390	482	-1392
-1570	-1386	580	-1395
-1470	-1390	684	-1395
-1375	-1388	779	-1391
-1276	-1390	874	-1390
-1179	-1386	975	-1391
-1071	-1386	1079	-1386
-972	-1389	1182	-1386
-873	-1394	1279	-1386
-774	-1395	1383	-1387
-673	-1400	1481	-1388
-570	-1393	1582	-1383
-476	-1390	1683	-1383
-372	-1390	1784	-1387
-269	-1391	1882	-1391
-170	-1387	1984	-1390
-71	-1388	2086	-1386
71	-1387	2185	-1385
170	-1391	2286	-1389
270	-1393	2385	-1387
377	-1392		

Abbildung A.1.: Position der einzelnen Messpunkte erfasst mittels Trackingsystem

A.3. Daten der Probanden

Proband 1		
	Ereignis	Auffälligkeiten
Test 1	links oben nach rechts unten	
Test 2	rechts unten nach links unten	
Test 3	rechts unten nach oben links	
Test 4	rechts mittig nach links mittig	schwer zu lokalisieren
Test 5	links oben nach rechts oben	schwer zu lokalisieren
Test 6	links unten nach rechts unten	schwer zu lokalisieren
Test 7	links oben nach rechts unten	
Test 8	rechts oben nach links oben	
Test 9	rechts unten nach links oben	
Test 10	rechts nach links mittig	schwer zu lokalisieren
Test 11	links unten nach rechts oben	schwer zu lokalisieren
Test 12	rechts unten nach links unten	schwer zu lokalisieren

A. Benutzertest und Messpunkte

Proband 2		
	Ereignis	Auffälligkeiten
Test 1	links oben nach rechts oben	zwischen durch leiser
Test 2	rechts nach links	
Test 3	rechts nach Mitte gleiche Höhe	
Test 4	rechts nach links	
Test 5	leicht links nach rechts nicht ganz links und rechts	
Test 6	links nach rechts	leichte Lautstärkeerhöhung bei Bewegung
Test 7	links oben nach rechts mittig	rechts kleine Lautstärke-schwankung
Test 8	rechts oben nach links oben	
Test 9	rechts mitte nach links oben	
Test 10	rechts oben nach links oben	
Test 11	links oben nach rechts oben	
Test 12	rechts nach links	schwer zu lokalisieren

Proband 3		
	Ereignis	Auffälligkeiten
Test 1	links oben nach rechts oben	Schwankung der Lautstärke
Test 2	mittig rechts nach links	
Test 3	rechts unten nach links oben	
Test 4	rechts oben nach links oben	schwer zu lokalisieren
Test 5	links unten nach rechts unten	schwer zu lokalisieren
Test 6	links unten nach rechts oben	schwer zu lokalisieren
Test 7	links oben nach rechts unten	Schwankung der Lautstärke
Test 8	mittig rechts nach links	
Test 9	rechts unten nach links oben	
Test 10	rechts oben nach links oben	schwer zu lokalisieren
Test 11	links nach rechts mittig	schwer zu lokalisieren
Test 12	rechts nach links	schwer zu lokalisieren

A. Benutzertest und Messpunkte

Proband 4		
	Ereignis	Auffälligkeiten
Test 1	links oben nach rechts unten	Lautstärke gleich, Lokalisation Mitte schlecht
Test 2	mittig rechts nach links	keine Horizontale Bewegung
Test 3	rechts unten nach links oben	Am Anfang und Ende deutlich Lauter
Test 4	rechts oben nach links unten	Lokalisation sehr schwammig, Lautstärke gleich
Test 5	links Mitte nach rechts Mitte	Lokalisation sehr schwammig, Lautstärke gleich
Test 6	links unten nach rechts oben	Lokalisation besser als bei 5
Test 7	links oben nach rechts Mitte	unten Hohe Töne
Test 8	rechts nach links eher oben	unten Hohe Töne
Test 9	rechts unten nach links unten	Lokalisation besser als bei 7
Test 10	rechts oben nach links unten	gut zu lokalisieren
Test 11	links nach rechts mittig	gut zu lokalisieren
Test 12	rechts unten nach links oben	gut zu lokalisieren. Lautstärke

Proband 5		
	Ereignis	Auffälligkeiten
Test 1	nicht Wahrnehmbar	Sprung in Wahrnehmung
Test 2	rechts nach links	
Test 3	rechts unten nach links oben	
Test 4	rechts oben nach links unten	
Test 5	links nach rechts	
Test 6	links unten nach rechts oben	
Test 7	links unten nach rechts	
Test 8	rechts nach links	
Test 9	rechts unten nach links oben	
Test 10	rechts unten nach links oben	
Test 11	links nach rechts	
Test 12	rechts unten nach links oben	

A. Benutzertest und Messpunkte

Proband 6		
	Ereignis	Auffälligkeiten
Test 1	links nach rechts Kopfhöhe	
Test 2	rechts nach links über Kopfhöhe	
Test 3	rechts nach links waagrecht	wird lauter
Test 4	rechts nach links waagrecht über Kopfhöhe	
Test 5	Mitte links nach Mitte rechts	Lokalisierung unscharf
Test 6	links nach rechts über Kopfhöhe	
Test 7	links oben nach rechts unten	Lokalisierung unscharf
Test 8	rechts oben nach links Mitte	Lokalisierung unscharf, Schwankung der Lautstärke
Test 9	rechts oben nach links unten	leichte Schwankung der Lautstärke
Test 10	rechts oben nach links oben	
Test 11	Mitte links nach Mitte rechts	Lokalisierung sehr unscharf
Test 12	Mitte rechts nach Mitte links	Lokalisierung unscharf

Proband 7		
	Ereignis	Auffälligkeiten
Test 1	links oben nach rechts unten	
Test 2	rechts oben nach links oben	
Test 3	rechts oben nach links oben	
Test 4	rechts oben nach links unten	Bewegung nicht linear
Test 5	links oben nach rechts oben	leichtes Rauschen, unnatürliches Echo
Test 6	links unten nach rechts oben	schriller Restton von links, als Ton schon rechts war
Test 7	halb links oben nach rechts oben	Lokalisierung unscharf
Test 8	rechts oben nach links oben	Eindruck von Schwankungen, als würde Objekt unter der rechts links Bewegung hin und her wackeln
Test 9	rechts unten nach links oben	Schwankung der Lautstärke
Test 10	rechts oben nach links unten	scheint zwischenzeitlich aus allen Richtungen zu kommen
Test 11	links oben nach rechts oben	
Test 12	rechts oben nach links oben	Lokalisierung sehr unscharf

Proband 8		
	Ereignis	Auffälligkeiten
Test 1	links oben nach rechts unten	leicht leiser
Test 2	rechts nach links	
Test 3	rechts oben nach links unten	wird lauter
Test 4	rechts nach links	
Test 5	links nach rechts	
Test 6	links unten nach rechts oben	schwer zu lokalisieren
Test 7	links oben nach rechts unten	wird leiser
Test 8	rechts nach links	
Test 9	rechts nach links	
Test 10	rechts oben nach links unten	schwer zu lokalisieren
Test 11	links nach rechts	
Test 12	rechts oben nach links oben	

B. Matlab-Code

B.1. Simulation

```
1
2 clear all
3
4 %% -----
5 %HIER DIE POSITION DER SOURCE UND MESSABSTAENDE AENDERN
6 positionSrc = [0,-2,2]; %Position der Quelle
7
8 %Messabstaende
9 messAbstandX = 0.01;
10 messAbstandY = 0.01;
11 %% -----
12
13 F = 1000; %Frequenz [Hz]
14 sSpeed = 343; % Schallgeschwindigkeit [m/s]
15 lambda = sSpeed / F; % Wellenlaenge [m]
16 positionHoerer = [0,0,0]; %Position des Messpunktes
17 twonderConfRef = 3.3; %nach config
18 dreiDNormal = [0, 1, 0]; %ausrichtung
19 transitionRadius = 0.1 * 1.5;
20
21 %Dimensionen der Vektoren
22 nSpeakers = 48;
23 %nSpeakers = 4; %fuer 4 Ls
24 %nSpeakers = 96; %fuer 96 Ls
25 xIdx = 1;
26 yIdx = 2;
27 zIdx = 3;
28
29 %Anfangskoordianten des ersten Segments, Abstand betraegt 0.1.
30 %bei jeder 8. Spur betraegt Abstand 0.1153
```

B. Matlab-Code

```
31 startx = -2.3883;
32 %startx = -1.2230;           %fuer 4 Ls
33 starty = 0.0;
34 startz = 2;
35 abstandSpuren = 0.1;
36 %startz = 0.2;             %fuer 4 Ls
37 %abstandSpuren = 0.8154;   %fuer 4 Ls
38 endOfTopArray = 49;       %Ende des oberen Arrays (Lautsprecher)
39
40 %% -----
41 %Berechnung der Messpositionen
42
43 %Start-/Endpositionen
44 startxPos = [-2,1,2];
45 endxPos = [2,1,2];
46
47 startyPos = [-2,1,2];
48 endyPos = [-2,5,2];
49
50 %Berechnung der Anzahl der Messpunkte
51 zeile = ((dist3D(startxPos-endxPos)) / messAbstandX)+1;
52 spalte = ((dist3D(startyPos-endyPos)) / messAbstandY)+1;
53
54 %Dimension des Arrays fuer die Messpunkte
55 dim = (zeile*spalte);
56
57 %Array welches die Messpunkte enthaelt
58 messPos = zeros(dim,3);
59
60 %Initiale Zuweisung der Startpunkte
61 sX = startxPos(xIdx);
62 sY = startxPos(yIdx);
63 sZ = startxPos(zIdx);
64
65 temp = 0;
66
67 %Schleife zum Fuellen des Arrays
68 for i = 1:1:(dim)
69
70     %wenn X-Messreihe berechnet, Schritt auf Y-Achse gehen und
```

B. Matlab-Code

```
71 %und naechste X-Messreihe berechnen
72 if temp == zeile
73     sX = startxPos(xIdx);
74     sY = sY + messAbstandY;
75     temp = 0;
76 end %if
77
78 %fuellen des Arrays mit Koordinaten
79 messPos(i,xIdx) = sX;
80 messPos(i,yIdx) = sY;
81 messPos(i,zIdx) = sZ;
82
83 sX = sX + messAbstandX;
84
85 temp = temp + 1;
86 end % for
87 %% -----
88
89 %Vektoren mit den Amplituden und den Koordinaten
90 audioSignale = zeros(nSpeakers,1);      %Signale pro Messpunkt
91 koordinatenLS = zeros(nSpeakers,3);     %Koordinaten der Spuren
92 d_i = zeros(nSpeakers,1);              %Dist. von Spur zu Messpunkt
93 phiDistanz = zeros(nSpeakers,1);       %Phasenverschiebung Distanz
94 amplDistanz = zeros(nSpeakers,1);      %Amplitudenabnahme Distanz
95 datenSchwingungen = zeros(nSpeakers,1); %Ltspr- Schwingungszeiger
96 datenLS_Messpos = zeros(nSpeakers,1);  %Schwingungszeiger Ltspr-Mp
97 ergebnisSignale = zeros(dim,1);       %Ergebnissignal pro Messpunkt
98
99 temp = 1;
100
101 %Schleife zum fuellen des Arrays fuer die Koordinaten der Spuren
102 for i = 1:1:nSpeakers
103
104     %% zwischen jedem LS-Segment ist der Abstand groesser als zwischen
105     %% den Spuren, daher alle 8 Durchlaeufer Abstand aendern
106     if (temp == 8) && (i < endOfTopArray)
107         %if (temp == 8)                    %fuer 96 Ls
108             abstandSpuren = 0.1153;
109             temp = 0;
110
```

B. Matlab-Code

```
111 %% fuer die unteren vier Lautsprecher. Diese erst berechnen, wenn
112 %% alle oberen Spuren berechnet wurden
113 elseif i == endOfTopArray
114     startx = -1.223;
115     %startx = -2.3883;           %fuer 96 Ls
116     starty = 0.0;
117     startz = 0.2;
118     abstandSpuren = 0.8154;
119     %abstandSpuren = 0.1;       %fuer 96 Ls
120
121 %% normaler Abstand zwischen Spuren
122 elseif (i < endOfTopArray)
123 %else                           %fuer 96 Ls
124     abstandSpuren = 0.1;
125     %abstandSpuren = 0.8154;    %fuer 4 Ls
126 end % if
127
128 %% fuellen des Arrays mit den Koordianten der Spuren
129 koordinatenLS(i,xIdx) = startx;
130 koordinatenLS(i,yIdx) = starty;
131 koordinatenLS(i,zIdx) = startz;
132
133 startx = startx + abstandSpuren;
134
135 temp = temp + 1;
136
137 end %for
138
139 %% Schleife zum Berechnen der Amplituden und Phasen fuer Spur
140 for iSpeaker = 1:1:nSpeakers
141
142
143     %Algorithmus aus source.cpp berechnet per Koordinaten
144     %der Schallquelle und der Wiedergabekanäle das
145     %resultierende Audiosignal jedes Wiedergabekanals.
146
147 spur = koordinatenLS(iSpeaker,:);
148 srcToSpkVec = spur - positionSrc;
149 normalProjection = dot(srcToSpkVec,dreiDNormal);
150 srcToSpkDistance = dist3D(srcToSpkVec);
```

```
151 cosphi = normalProjection / srcToSpkDistance;
152
153 amplitudeFactor = (sqrt(twonderConfRef /
154     (twonderConfRef + normalProjection))) *
155     ( cosphi / sqrt(srcToSpkDistance));
156
157 datenSchwingungen(iSpeaker) = amplitudeFactor *
158     exp(j * 2 * pi * mod(srcToSpkDistance / lambda, 1));
159
160 end % for
161
162 for k = 1:1:dim
163     %% holen der einzelnen Positionen an dem sich Hoerer(Mikro)
164     %% befindet
165     positionHoerer = messPos(k, :);
166
167     % Schleife zum berechnen der Distanz von Spur zu Hoerer
168     d_i = zeros(nSpeakers, 1);
169
170     for iSpeaker = 1:1:nSpeakers
171
172         %% Abstand von Audiospur zu Messposition
173         d_i(iSpeaker) = dist3D(koordinatenLS(iSpeaker, :)
174             - positionHoerer);
175
176     end % for
177
178     phiDistanz = 2 * pi * mod(d_i / lambda, 1);
179     datenLS_Messpos = 1 ./ d_i .* exp(j * phiDistanz);
180     ergebnisSignale(k) = sum(datenLS_Messpos .* datenSchwingungen);
181
182 end %for
183
184 plotMatrix = reshape (ergebnisSignale, zeile, spalte);
185 plotMatrix = (abs(plotMatrix).^2)';
186
187 imagesc(-2:messAbstandX:2, 1:messAbstandY:5, plotMatrix);
```

Listing B.1: Matlab Skript zur Simulation der Schallfelder

```
1
2 function distance = dist3D( v )
3
4 %% distance = dist3D(vec3D)
5 %% calculate Euclidean norm of 3D-vector v
6 %% i.e., sqrt(v_x^2 + v_y^2 + v_z^2)
7
8 %% Sanity checks
9 sz = size(v);
10 if sz(2) ~= 3 || sz(1) ~= 1
11     error ('dist3D expects a 3-column Matrix');
12 end % if
13
14 distance = sqrt(v(1)^2 + v(2)^2 + v(3)^2);
15
16 end
```

Listing B.2: Matlab Hilfsskript zur Berechnung des Betrags eines Vektors

B.2. Auswertung Messdaten

```
1
2 clear all
3
4 mr = 4;      %Anzahl Messreihen
5
6 %Platzhalter
7 mr1 = 1;
8 mr2 = 2;
9 mr3 = 3;
10 mr4 = 4;
11
12 rCell = 1;
13 lagsCell = 2;
14
15 %Einlesen der Messdaten
16 wavFilesMr1 = dir('Messreihe1*.wav');
17 wavFilesMr2 = dir('Messreihe2*.wav');
18 wavFilesMr3 = dir('Messreihe3*.wav');
```

```
19 wavFilesMr4 = dir('Messreihe4*.wav');
20
21 %Anzahl der Messdaten
22 lengthMr = length(wavFilesMr1);
23
24 %Matrizen der Messdaten
25 messDaten = cell(mr, lengthMr);           %Rohdaten
26
27 %Kreuzkorrelationen der einzelnen Messreihen
28 kreuzKorrMr1 = cell(2, lengthMr);
29 kreuzKorrMr2 = cell(2, lengthMr);
30 kreuzKorrMr3 = cell(2, lengthMr);
31 kreuzKorrMr4 = cell(2, lengthMr);
32
33 %Matrix mit positiven Anteilen der korrelierten Messwerte
34 rng = cell(mr, lengthMr);
35
36 %Ergebnisintensitäten, quadriert und aufsummiert
37 ergIntens = zeros(mr, lengthMr);
38
39 %Einlesen des Inputsignals
40 inputSignal = wavread('whitenoise10sec.wav');
41
42 %Einlesen der Rohdaten
43 for k = 1:1:lengthMr
44     messDaten{mr1,k} = wavread(wavFilesMr1(k).name);
45     messDaten{mr2,k} = wavread(wavFilesMr2(k).name);
46     messDaten{mr3,k} = wavread(wavFilesMr3(k).name);
47     messDaten{mr4,k} = wavread(wavFilesMr4(k).name);
48 end
49
50 %-----
51 %Kreuzkorrelationen der einzelnen Messreihen
52 for k = 1:1:lengthMr
53     [r, lags] = xcorr(messDaten{mr1,k},inputSignal);
54     kreuzKorrMr1{rCell,k} = r;
55     kreuzKorrMr1{lagsCell,k} = lags;
56 end
57
58 for k = 1:1:lengthMr
```

```
59     [r, lags] = xcorr(messDaten{mr2,k},inputSignal);
60     kreuzKorrMr2{rCell,k} = r;
61     kreuzKorrMr2{lagsCell,k} = lags;
62 end
63
64 for k = 1:1:lengthMr
65     [r, lags] = xcorr(messDaten{mr3,k},inputSignal);
66     kreuzKorrMr3{rCell,k} = r;
67     kreuzKorrMr3{lagsCell,k} = lags;
68 end
69
70 for k = 1:1:lengthMr
71     [r, lags] = xcorr(messDaten{mr4,k},inputSignal);
72     kreuzKorrMr4{rCell,k} = r;
73     kreuzKorrMr4{lagsCell,k} = lags;
74 end
75 %-----
76
77 %finden der Positiven Lags der Kreuzkorrelierten
78 for k = 1:1:lengthMr
79     rng{mr1,k} = find(kreuzKorrMr1{lagsCell,k} >= 0);
80     rng{mr2,k} = find(kreuzKorrMr2{lagsCell,k} >= 0);
81     rng{mr3,k} = find(kreuzKorrMr3{lagsCell,k} >= 0);
82     rng{mr4,k} = find(kreuzKorrMr4{lagsCell,k} >= 0);
83 end
84
85 %Berechnung der Intensitäten durch quadrieren und aufsummieren der
86 %positiven Lags
87 for k = 1:1:lengthMr
88
89     r1tmp = kreuzKorrMr1{rCell,k};
90     r2tmp = kreuzKorrMr2{rCell,k};
91     r3tmp = kreuzKorrMr3{rCell,k};
92     r4tmp = kreuzKorrMr4{rCell,k};
93
94     rng1 = rng{mr1,k};
95     rng2 = rng{mr2,k};
96     rng3 = rng{mr3,k};
97     rng4 = rng{mr4,k};
98
```



```
99   quad1 = r1tmp(rng1).^2;
100  quad2 = r2tmp(rng2).^2;
101  quad3 = r3tmp(rng3).^2;
102  quad4 = r4tmp(rng4).^2;
103
104  ergIntens(mr1,k) = sum(quad1);
105  ergIntens(mr2,k) = sum(quad2);
106  ergIntens(mr3,k) = sum(quad3);
107  ergIntens(mr4,k) = sum(quad4);
108
109  end
110
111  %Plot
112
113  startx = -2;
114  starty = 1;
115
116  endx = 2;
117  endy = 4;
118
119  imagesc(startx:0.1:endx,starty:1:endy,ergIntens);
```

Listing B.3: Matlab Skript zur Auswertung der Messdaten

Literaturverzeichnis

- [Ahr12] Jens Ahrens. *Analytic Methods of Sound Field Synthesis*. T-Labs Series in Telecommunication Services. Springer-Verlag, Berlin Heidelberg, 2012.
- [Aud] Synthetic Wave Audio. Virtuelle Schallquellen. http://www.syntheticwave.de/virtuelle_Schallquelle.htm. Zugriff: 02-02-2015.
- [Baa07] Marije A.J. Baalman. On wave field synthesis and electro-acoustic music, with a particular focus on the reproduction of arbitrarily shaped sound sources. Master's thesis, TU Berlin, 2007.
- [BdVV93] A. J. Berkhout, D. de Vries, and P. Vogel. Acoustic control by wave field synthesis. *Journal of the Acoustical Society of America Vol. 93*, 1993.
- [BHSK07] Marije A.J. Baalman, Torben Hohn, Simon Schampijer, and Thilo Koch. Renewed architecture of the swonder software for wave field synthesis on large scale systems. *LAC2007 - 5th International Linux Audio Conference*, 2007.
- [Chr14] Carola Christoffel. Modifikation der Software einer Wellenfeldsyntheseanlage zur Wiedergabe fokussierter Quellen in Abhängigkeit der Zuhörerposition. Bachelorarbeit, Hochschule für Angewandte Wissenschaften Hamburg, 2014.
- [Foh13] Wolfgang Fohl. The wave field synthesis system at the haw hamburg. In *Sound - Perception - Performance Bd. 1*. Springer. Bader, Rolf (Hrsg.), 2013.
- [Föll] Helmut Föll. Einführung in die Materialwissenschaft II. http://www.tf.uni-kiel.de/matwis/amat/mw2_ge/kap_3/backbone/r3_4_1.html#_7. Zugriff: 10-02-2015.
- [Gör08] Thomas Görne. *Tontechnik*. Hanser Fachbuch, 2008.
- [Han14] Martin Hansen. Positionierung von Klangquellen einer Wellenfeldsynthese-Anlage mit Hilfe eines Audio-Plugins. Bachelorarbeit, Hochschule für Angewandte Wissenschaften Hamburg, 2014.

- [JD03] Sebastien Moreau Jerome Daniel, Rozenn Nicol. Further Investigations of High Order Ambisonics and Wavefield Synthesis for Holophonic Sound Imaging. *Audio Engineering Society Convention Paper 5788*, 2003.
- [MGTK12] Michael Makarski, Anselm Goertz, Rainer Thaden, and Jochen Kleber. Entwicklung einer WFS-Anlage mit FIR-Entzerrung und Dante Audionetzwerk. Technical report, Institut für Akustik und Audiotechnik and Four Audio, 2012.
- [Mon11] Matthew N. Montag. Wave field synthesis in three dimensions by multiple line arrays. Master's thesis, University of Miami, 2011.
- [Nog12] Malte Nogalski. Gestengesteuerte Positionierung von Klangquellen einer Wellenfeldsynthese-Anlage mit Hilfe eines kamerabasierten 3D-Tracking-Systems. Bachelorarbeit, Hochschule für Angewandte Wissenschaften Hamburg, 2012.
- [Sen] Eberhard Sengpiel. Abnahme des Schallpegels in Dezibel in Abhängigkeit von der Änderung der Entfernung. <http://www.sengpielaudio.com/Rechner-entfernung.htm>. Zugriff: 01-04-2015.
- [The80] Günther Theile. *Über die lokalisation im überlagerten Schallfeld*. PhD thesis, TU Berlin, 1980.
- [The04] Günther Theile. Wave field synthesis - a promising spatial audio rendering concept. *7th Int. Conference on Digital Audio Effects*, 2004.
- [WFM03] Matthew Wright, Adrian Freed, and Ali Momeni. Opensound control: State of the art 2003. *Conference on New Interfaces for Musical Expression*, 2003.
- [Wri02] Matthew Wright. The open sound control 1.0 specification. http://opensoundcontrol.org/spec-1_0, 2002. Zugriff: 26-01-2015.

Hiermit versichere ich, dass ich die vorliegende Arbeit ohne fremde Hilfe selbständig verfasst und nur die angegebenen Hilfsmittel benutzt habe.

Hamburg, 15. Mai 2015

Marvin Geitner