On Wave Field Synthesis and electro-acoustic music, with a particular focus on the reproduction of arbitrarily shaped sound sources

vorgelegt von
Marije A.J. Baalman
Master of Science
aus Pingjum

Von der Fakultät I - Geisteswissenschaften
der Technischen Universität Berlin
zur Erlangung des akademischen Grades
Doktor der Philosophie
Dr. phil.

genehmigte Dissertation

Promotionsausschuss:
Vorsitzender: Prof. Dr. Peter Erdmann
Berichter: Prof. Dr. Stefan Weinzierl
Berichter: Prof. Dr. Diemer de Vries

Tag der wissenschaftliche Aussprache: 21. Dezember 2007

Berlin 2008

D 83

Summary

Spatial sound has long been a topic of interest in electro-acoustic music. With continual technical advancements of methods to achieve spatialisation of sound, an increase in the realisation of spatial sound ideas can be achieved, while concurrently giving birth to new concepts in this area. Wave Field Synthesis (WFS) provides a compelling method, as it excels in creating a clear localisation of a virtual sound source for a large listening area. Despite the necessary hardware investment, WFS is gaining popularity at many institutes - internationally - providing increasing numbers of composers and sound artists with access to the technology. This relationship between technology and composer provides an environment for exploration of strengths and weaknesses, as well as driving forward development of the technology while composers are creating the content. This can be considered a key element in creating a demand for WFS in commercial and academic markets.

To provide more extensive control over sound sources, this thesis proposes a method for the reproduction of arbitrarily shaped objects emitting sound, using WFS. The method begins with a known or defined geometry of the object surface, and assumes a known or defined vibration of this surface. Using these parameters, signals for a WFS loudspeaker array are calculated, applying an adapted WFS operator that takes into account the elevation above or below the horizontal plane, and diffraction of the sound around the object itself.

The method is validated through the calculation of several examples, evaluating the effects of size, shape, refinement, distance and diffraction on the resulting wave field emitted by the loudspeakers. The diffraction model gives plausible results, but a "second stage" of diffraction occurs at the WFS reproduction array. A listening test shows that sound objects reproduced in this way cause significant perceptual effects primarily on tone colouration, but also in various spatial parameters such as source width, localisation and spaciousness.

Zusammenfassung

Seit Beginn der elektroakustischen Musik stößt das Thema des räumlichen Klanges unter Komponisten auf großes Interesse. Durch die kontinuierliche technische Entwicklung konnten und können diesbezüglich zunehmend Methoden zur Realisierung von Raumklang erreicht und neue Konzepte angeregt werden. Die Wellenfeldsynthese (WFS) zeichnet eine gewichtige Methode aus, die eine klare Lokalisation von virtuellen Klangquellen für einen großen Hörraum kreiert. Trotz des hierfür notwendigen hohen technischen und finanziellen Aufwands gewinnt die Wellenfeldsynthese international an vielen Instituten an Popularität, wodurch Komponisten und Klangkünstlern in zunehmendem Maße Zugang zu dieser Technologie ermöglicht wird. In dieser Beziehung zwischen Technologie und K"unstlern bietet sich die Möglichkeit der Erforschung der Stärken und Schwächen, ebenso wie der Weiterentwicklung der Technologie bei gleichzeitigem Herstellen von Inhalten durch die Künstler. Dieses ist auch der Schlüssel des Erfolges, um die Nachfrage für die WFS anzuregen und ihr einen größeren Markt zu erschließen.

Um eine größere und detailliertere Kontrolle über Klangquellen liefern zu können, wird in dieser Dissertation eine Methode vorgestellt für die Reproduktion (mittels WFS) von willkürlich geformten Klang-aussendenden Objekten. Diese Methode geht von einer bekannten, beziehungsweise einer definierten Geometrie der Oberfläche eines Objektes aus und nimmt eine bekannte oder definierte Schwingung dieser Oberfläche an. Mit diesen Parametern werden Signale für ein WFS-Lautsprecher-Array, durch die Anwendung eines angepassten WFS-Operator berechnet, welcher auf die Position über und unter der horizontalen Ebene sowie die Beugung des Klangs um das Objekt herum Bezug nimmt.

Diese Methode wird durch die Kalkulation einiger Beispiele überprüft, wobei die Effekte von Größe-, Form-, Verfeinerungs-, Distanz- und Beugungsparametern auf das resultierende von den Lautsprechern wiedergegebene Wellenfeld ausgewertet werden. Das Beugungsmodell ergibt plausibele Resultate; es tritt aber ein "zweite Stufe" der Beugung auf, auf dem Lautsprecher-Array.

Ein Hörtest hat deutlich gezeigt, das die Schallquellen, die auf diese Weise wiedergegeben wurden, deutlich wahrnehmbare Eigenschaften aufweisen. Als Erstes ist ein Effekt auf die Klangfarbe zu hören, aber des weiteren auch in verschiedenen räumlichen Parametern wie Quellenbreite, Lokalisation und Räumlichkeit.

Samenvatting

Ruimtelijk geluid is al sinds de begintijden van de electro-akoestische muziek een thema dat grote interesse wekt onder componisten. Met de voortdurende technische ontwikkelingen op het gebied van ruimtelijk geluid, vinden ook steeds meer ideeën voor ruimtelijkheid een uitdrukking, terwijl tegelijkertijd de nieuwe technische mogelijkheden nieuwe ideeën inspireren. Golfveldsynthese (Wave Field Synthesis, WFS) biedt een fascinerende methode, die uitblinkt in het creëren van een kristalheldere localisatie van een virtuele geluidsbron, welke op een groot luisteroppervlak hoorbaar is. Ondanks de benodigde investeringen in hardware wint WFS aan populariteit bij vele instituten - op internationaal vlak - en voorziet daarmee meer en meer componisten en geluidskunstenaars van toegang tot de technologie. Deze relatie tussen techniek en componist biedt een omgeving voor het verkennen van de sterke en zwakke kanten van de technologie, en is ook een drijvende kracht voor verdere ontwikkeling van de technologie, terwijl tegelijkertijd componisten inhouden creëren. Dit kan gezien worden als een sleutelelement om een vraag naar WFS in commerciële en academische markten tot stand te brengen.

Om in een grotere en gedetailleerde beheersing over geluidsbronnen te voorzien, wordt in deze dissertatie een methode voorgesteld voor de reproductie, met behulp van WFS, van willekeurig gevormde objecten die klank voortbrengen. De methode gaat uit van een bekende of gedefiniëerde geometrie van de oppervlakte van het object, en neemt een bekende of gedefiniëerde trilling van deze oppervlakte aan. Met deze parameters worden signalen voor een WFS luidspreker array berekend, met behulp van een aangepaste WFS operator, die de verticale component, boven of onder het horizontale vlak, in rekening neemt, en diffractie van het geluid door het object zelf.

De methode wordt gevalideerd door de berekening van verscheidene voorbeelden, die het effect van grootte, vorm, discretisatie, afstand en diffractie op het gereproduceerde golfveld laten zien. Het diffractiemodel levert plausible resultaten op in deze toepassing; er treedt echter een “tweede stap” voor diffractie op aan het luidsprekerarray. Een luistertest heeft laten zien dat geluidsbronnen die op deze manier weergegeven worden, duidelijke perceptieve eigenschappen hebben; in de eerste plaats is er een effect op de klankkleur, de andere eigenschappen zijn van ruimtelijke aard en hebben betrekking op de bronbreedte (source width), lokalisatie en ruimtelijkheid.

Preface

This thesis is the result of occupying myself with Wave Field Synthesis for approximately the last five years.

My first interest in Wave Field Synthesis was born by the very first demonstration of WFS I experienced, as I studied Applied Physics in Delft. Though I originally intended to do my Master's Thesis on the topic, this did not work out due to circumstances. While I was working on my Master's Thesis on Spaciousness in Concert Halls, my supervisor Diemer de Vries, was an Edgard Varèse guest professor at the TU Berlin, and upon his return to Delft, he held a presentation about his time in Berlin. Being halfway the one year Sonology Course at the Conservatory in The Hague, I asked him, approximately one week before I handed in my Master's Thesis whether he thought it would be possible for me to go to Berlin, to work on Wave Field Synthesis and electro-acoustic music; he put me in contact with Folkmar Hein, who immediately showed enthusiasm, but also clearly stated the reality that there was no financial support to be offered from the side of the TU Berlin at that moment. Thus I ventured into the world of grant applications, and have been extremely lucky in receiving grants from the VSB Fonds (for a 10 month post graduate research project) and the DAAD (for a 6 month post graduate research project), and have been lucky to have supportive parents in the months where grant money was not available, as well as to have found a temporary job at the neighbouring Institut für Technische Akustik (ITA).

With the appointment of Stefan Weinzierl as the new professor for the Fachgebiet in 2004, he convinced me to expand on the theme of arbitrarily shaped sound sources as a Ph.D. research project, which has been funded by a grant of the *NaFöG* (Nachwuchsförderungsgesetz) in Berlin (here I need to thank Frau Susanne Hördt for the great care with which she handled all the details of the grant).

Upon commencing to actually write down the thesis (partly because in the same period, the TU Berlin has been able to realise a WFS system in one of its lecture halls) it appeared best to combine both the work on the software and its renewed version, as well as its application by composers, and the research into arbitrarily shaped sound sources, into one thesis, especially as this puts the thesis not only within a physical context (which is the main emphasis of the arbitrarily shaped sources), but also into a more musical context, which is more fitting to a faculty of "Geisteswissenschaften", to which our Audio Communi-

cation Group belongs. It also gave me an opportunity to elaborate on many topics, which have not been described in many other WFS theses.

This is the place to thank everyone who has supported me over the years in this work, which not only includes my supervisors Stefan Weinzierl and Diemer de Vries, but also Folkmar Hein, who has been a great support throughout my five years at the TU, the various students and tutors in the studio: Hanns Holger Rutz (who introduced me into the underground electronic music scene of Berlin), Niklas Werner (†), Martin Rumori and Stefan Kersten (who all helped me get started with Linux Audio), and of course, the slowly expanding WFS team who rewrote the software and set up the system in the lecture hall: Torben Hohn, Simon Schampijer (who has also been a great co-organiser of the Linux Audio Conference 2007), Daniel Plewe, Eddie Mond and Thilo Koch, as well as Wilm Thoben, Eckie Güther, Florian Goltz and Sebastian Roos. Hans-Joachim Maempel and Judith Liebetrau have given valuable advice on the listening tests. Thanks go to Somaya Langley for proofreading and editing my English for parts of this thesis. Julius Stahl and Ilka Theurich have helped me create the German version of the summary. My brother Philippus read my whole thesis and found typos and sentences which weren't making any sense, so I could correct them for the final publication.

For inspiration I have to thank several of the Edgard Varèse guest professors, besides Diemer, Hans Tutschku for inspiration and the first experiments combining Modalys and WFS, and Norbert Schnell for inspiration for realisation of the spatial granular synthesis.

Outline of this thesis

The first chapter gives an introduction to spatial techniques as used in electro-acoustic music and goes into detail on spatial perception of source shape and size. Chapter 2 explains the theory of wave field synthesis, whereas chapter 3 details the setup of hardware and software for WFS as developed at the TU Berlin, focusing on the practical details of implementation. The chapter that follows will discuss the use of Wave Field Synthesis for electro-acoustic music.

In chapter 5, a source model for 3D sounding objects is introduced, including diffraction and local reflections of the source object. Chapter 6 details the algorithms used in the software implementation of this model. Chapter 7 presents an evaluation of the model.

The thesis is concluded with a discussion of the results and an outlook to the future of WFS.

Publications

Parts of the work presented in this thesis have been published previously in the following conference papers. Several of the results presented in these papers have been reviewed and refined for this thesis.

Reviewed conference papers

Marije A.J. Baalman. On wave field synthesis and electro-acoustic music - State of the Art 2007. In *International Computer Music Conference 2007, Copenhagen, Denmark, 27-31 August 2007*, 2007.

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. In *Linux Audio Conference 2007, TU Berlin, 22-25 March 2007*, 2007.

M.A.J. Baalman. *swonder3Dq*: Auralisation of 3D objects with wave field synthesis. In *4th International Linux Audio Conference, April 27-30, 2006, ZKM, Karlsruhe*, 2006.

M.A.J. Baalman. Elevation problems in the auralisation of sound sources with arbitrary shape with wave field synthesis. In *International Computer Music Conference 2005, 1-6 September 2005, Barcelona, Spain*, 2005.

M.A.J. Baalman. Updates of the WONDER software interface for using wave field synthesis. In *3rd International Linux Audio Conference, April 21-24, 2005, ZKM, Karlsruhe*, 2005.

M.A.J. Baalman and D. Plewe. WONDER - a software interface for the application of wave field synthesis in electronic music and interactive sound installations. In *International Computer Music Conference 2004, Miami, 1-6 November 2004*, 2004.

M.A.J. Baalman. Application of wave field synthesis in the composition of electronic music. In *International Computer Music Conference 2003, Singapore, 29 September - 4 October 2003*, 2003.

Invited conference papers

M.A.J. Baalman. How to Control 840 Channels. Working with Large Scale Wave Field Synthesis. in *Next Generation, ZKM, Karlsruhe*, June 2007. `http://www.zkm.de`

M.A.J. Baalman. Discretisation of complex sound sources for reproduction with wave field synthesis. In *DAGA '05, 14 - 17 March 2005, München*, 2005.

M.A.J. Baalman. Toepassing van golfveldsynthese in de compositie van electronische muziek, In *NAG Journaal, bijeenkomst van 17 september 2003.* Nederlands Akoestisch Genootschap, 2003.

Other conference papers

Marije A.J. Baalman. Reproduction of arbitrarily shaped sound sources with wave field synthesis - physical and perceptive effects. In *AES 122nd Convention, Vienna, 5-8 May 2007*, 2007.

M.A.J. Baalman. Reproduction of arbitrarily shaped sound sources with wave field synthesis - theory and implementation. In *Tonmeistertagung 2006, Leipzig, 16-19 November 2006*, 2006.

M.A.J. Baalman. Application of wave field synthesis in electronic music and sound installations. In *2nd International Linux Audio Conference, 29 april - 2 Mai 2004, ZKM, Karlsruhe*, 2004.

Online journals

M.A.J. Baalman. Application of Wave Field Synthesis in electronic music and sound art. In *The IDEA #7*, 2004. `http://retiary.net/idea/idea7/idea_7/marijeb/marije.htm`

Contents

Chapter 1

Introduction

Spatialisation has long been a topic of interest within electro-acoustic music[1] and consequently many composers, sound artists and authors have written about different approaches, and used different techniques and technologies. Wave Field Synthesis (WFS) fits into this tradition and provides the possibility to realise many ideas which were previously impossible.

1.1 Compositional approaches

1.1.1 Creating space and spatiality

Raaijmakers [Raa93] states that throughout the 19th century the world has been reduced from a round, full spatial world to a single point, as the factors path (to be understood as process) and time were gained, in exchange for space, equated to soul. Since then, attempts have been made to regain lost space, through (1) technical, (2) spiritual and utopian means and (3) through art: by composing space.

To the first point, Raaijmakers ascribed the development from monophonic, via stereophonic, quadraphonic, ambisonic, to "spacesound", omniversum and holophony. To the second point, he provides an overview of how several artists and theorists imagine new spatiality - ones that cannot yet be realised - before continuing on to address, at greater extent, the topic of composed space. Beginning this, is a schematic listing of the relation between music transmission and spatial setting of the performance:

1. the lonely singer in an empty world;
2. contra-point, duo, dialogue, duel;
3. circus, camp-fire, jail;

[1] With electro-acoustic music is referred to music that is created by synthesis or processing, digitally and/or electronically, which is dependent on the reproduction by an electro-mechanical transducer, e.g. a loudspeaker. This specifically excludes the use of processing audio data with the aim to reproduce an acoustic event (e.g. a violin concert). It does include, however, any genre of electronic music, as well as (most) sound installations.

4. amphitheatre (as an in-between form);
5. 19th century concert hall, spaciousness, stereo, depth, hierarchy;
6. pop-screen, 1D, electronic, no depth, artificial space, laser/smoke, colour/light.

Modern composers do not desire space to passively "reproduce" music, but wish for it to become an active "source", so as to mobilise it. Three ideals can be used to achieve this ideal:

1. "Traces", "routes" and "planes" are set out in the space, that the sound can follow. With the aid of loudspeakers, imaginary sonic routes are created and movements are simulated. Forms of antiphony and sonic polyphony are created. Examples can be found in the works of Stockhausen, Varèse, Xenakis, Nono and Boerman. The sounding space is autonomous and independent of the topographical space in which the sounding is taking place.
2. Space (or location) is seen as a serial factor in the compositional process, equivalent to other parameters such as pitch, duration, loudness and tone colour, as done by Goeyvaerts and Stockhausen.
3. Development of a "composed acoustic" by the principle of auto summation of sound, aiming to transcend architectural acoustics by the "composed acoustics". The latter being subjected to compositional laws (e.g. the method of Boulez in Répons).

Joel Ryan[2] considers that space can be addressed in two ways: the first is to create paths for the sounds, resulting in a choreography of the sounds, and consequently, this choreography requires meaning to be inserted. An alternative way to address space is by attempting to fill the space with sound (which he often does in his own work) and can be achieved by using techniques such as reverb and delay.

1.1.2 Symbolic meanings of space and context

Trochimczyk [Tro01] relates the use of different spatial designs closely to the symbolic meaning of these forms, to spatial meanings encountered in our culture. Distinguishing between circles and nets, the first is associated with the familiar, but at the same time to the mystical and centralised, whereas the second is associated with a more democratic concept.

At the start of the discussion in her paper, she states that there is no such thing as spatial music, but only spatio-temporal music, as we can only experience space in time, and in turn, can only experience time in space.

Like other composers and authors (e.g. Boerman [Boe92] and Böhmer [B9̈3]), she elaborates on the issue of mobility of the audience: the music is no longer composed for an audience who are fixed in their listening position, but rather the notion is to permit audiences to move, creating their own personal experience of a certain piece, rather than a (spatially) fixed experience. This concept is

[2] source: a private conversation with the author

partly situated in a socio-political context, where the composer can no longer dictate the auditory experience of the listener, but rather the listener takes on an active role in his or her own experience, gaining freedom that is driven by his or her own curiosity. Practical concerns arise, such as that of the audience making noises themselves whilst moving (which can be addressed by carpeting spaces so footsteps are not heard [B9̈3]), as well as being distracted away from focussed listening by having to concentrate on appropriate movement through the space, to cite Xenakis [Xen94]:

> "The problem is that when people move around they cannot listen in the same way, they do not concentrate on the music. They do not know how to pay attention while walking and they do not notice the fact that when they change position they have a different aural perspective caused by the difference in location. Besides, they are distractive, they annoy other people who are listening, when they move."

Opposing this is Cage [Cag67], in his description of the behaviour of the audience in *Diary: Audience 1966*:

> "An audience can sit quietly or make noises. People can whisper, talk, and even shout. An audience can sit still or it can get up and move around. People are people, not plants."

Xenakis' approach in *Terretektorh* and *Nomos Gamma* is to disperse the orchestra between the audience, so spatial trajectories can be created by sequentially allowing the various instrumentalists to play the sounds; he also introduces an intimacy between the audience and the sound. As Boulez notes about *Répons* [Tro01], if an audience member wants to hear a different perspective on the piece, he or she can listen to it again from another location. Of course, this means that an audience should be able to have access to several performances, but in reality this is rarely the case.

An major inhibitive factor for creating music for mobile audiences, is that concert halls are often not suitable for such situations: due to social traditions, layout (where seats are fixed), plus security concerns. In contrast, in a non-classical context (popular music, club venues or underground/experimental music), the audience is more likely to be able to move around the space, changing their perspective. A good example of this are the concerts of the Dutch band *Kong* from the 1990's, where they performed in a quadraphonic layout. Each of the four musicians played on a small stage placed on each side of the hall and the audience was able to freely move (and dance) in between them.

Mobility of audiences can be more easily achieved in sound installations [dlMH99], which are usually presented in gallery or museum environments, and that are often open to the audience over extended time frames. Listeners can choose when to enter and when to leave, and are provided with the opportunity to shift their position in the space. This context introduces the problem where compositions intended for this space must be composed in a way that is suitable for this type of presentation. Commonly found, are looped compositions, although increasingly, generative compositions are composed for such spaces [Itu07]. When compositions are created to be responsive to action and moment inside the space (by the use of sensor technology), the process becomes directly related to the static architectural space, but also, and perhaps more importantly, to the dynamic social architecture of the space, i.e. the influence of the people present in the space and their attribution to environmental changes in each specific situation. As the sonic environment is dynamic, reacting in one way or another to individuals in the space, perception of the space is constantly changing.

Another form for spatial music is the mobility of the performers. Here the problem is that this very mobility may introduce a theatrical character to the work, which must be carefully evaluated. When the importance of the musicians' actions overshadows that of the sonorous results of these actions, music is transformed into musical theatre [Tro01].

1.1.3 Illusion, allusion and temporal development

Barrett [Bar02] gives an overview of composition strategies for spatial sound, discerning four main approaches to space and how the composer can work within it:

1. illusion of a space or a spatial location of an object,
2. allusion to a space or a spatial location of an object,
3. simulation of the three-dimensional sound field,
4. spatial possibilities contingent upon temporal development.

In the first case it is attempted to create a perceived space, which matches (as closely as possible) a real space. This is carried out by maintaining spatial laws such as the effect of sound transmission, the properties of the reverberant field, object image and multiple objects relationships, Doppler shifts and gestural-spatial definition. To create an effective spatial illusion, the technical implementation plays an important role. However, it is often sufficient to create a plausible spatial illusion via fairly simple means.

In the second case, space is implied by making use of listener's assumptions about certain sounds, e.g. through information that does not conform to the spatial laws, where insufficient spatial laws are present, or where presented spatial laws are set in conflict. An example of information other than the spatial laws is the sonic result of wind swaying a large tree in full leaf. When a listener hears this sound, he/she will assume an outdoor space.

In third case, the aim is to create an accurate simulation of a 3D sound field, rather than creating the illusion of one. This method is not based on creating psycho-acoustic illusions, but to create a physically accurate acoustical wave field. For a composer this demands a different working method, as it is object oriented, rather than track oriented.

The fourth case is where the spatial perception relies on the memory of the listener, be it from outside the composition (long-term memories, outerreferential development) or within the composition (self-referential development). Contrasts or paradoxes can be achieved by changing from a spatial "illusion" to an "allusion", or by changing from accurate spatial (and spectral) cues, to conflicting cues. One example of this is by moving a sound source quickly through the space and placing the Doppler shift at a position that is too early or too late. If the deviation from the natural physics is not too large, the listener will not perceive it as a breakdown of the spatial image, but rather perceive it as a transformation of the sounding object.

1.1.4 A terminology for spatial properties

Kendall [Ken07] presents a terminology of spatial attributes for describing spatial relationships of electro-acoustic music, beginning with a terminology of spatial attributes, as presented by Rumsey [Rum02], for reproduced sound. He uses the dimensional attributes of width, distance, depth and direction, as well as the immersive attributes of envelopment and presence. These elements can be applied on different levels of an auditory scene, from one source to groups of sources (such as an ensemble), to groups of groups, where the transition is continuous rather than discrete. As many levels as required in the composition being studied, can be used.

He also mentions four different frames of reference for "source":

1. the source signal, i.e. the acoustical signal or representation thereof;
2. the 'source image', i.e. the 'source' that has spatial attributes in the auditory scene;
3. the 'conceptual source', i.e. the object that the listener identifies with the sound, independent of the spatial attributes;
4. the listener's spatial schemata SOURCE (see [Joh87]).

1.1.5 Psycho-acoustical aspects

Our directional hearing serves a very primitive function, that of a warning system [Moo04, e.g.]. When we hear a sound, we need to be able to quickly identify the location of the sound, and whether the source producing the sound is a danger to us.

As the majority of events in the physical world occur within the horizontal plane, we have well developed capabilities for determining the position of a sound source in this plane. Also, we are better at localising sounds in front of us, than behind us [Bla97].

Our vertical perception is less developed. Our ability for horizontal localisation of sounds is based upon the fact that we have two ears, and are able to evaluate the differences in the signals we receive with each ear[3]. Whereas, localisation in the vertical plane is associated with spectral differences, due to the way sounds are reflected in our outer ear.

[3] front-back confusion can occur as the interaural differences for front and back may the same. Head movements help to avoid the confusion.

The main purpose for aural localisation is to enable us to direct our visual attention to the perceived sound source, which then determines our subsequent behaviour (such as seeking cover, moving out of the way, or aiming our arrows).

Sounds that occur behind us can make us feel uncomfortable, as we are often not able to distinguish them as quickly as sounds occurring in front of us.

It should also be noted that generally, we perceive sounds as coming from certain angles, and having a certain distance, i.e. we do not interpret the space in a Cartesian, but rather in a spherical coordinate system. Distance is directly related to feelings of intimacy: a close whisper in the ear, versus a distant shout. Sudden intrusions of the intimate sphere may cause discomfort, whereas a careful approach may be perceived as comforting.

Spatial dislocation of sources can also help us to perceive different streams of audio (stream segregation); our auditory system is capable of focusing our attention to a specific stream of audio, based on its location in space, and thus distinguishing the stream from another auditory stream. Musically, this gives the opportunity to use different sound source locations for different voices in the music (an example can be found in a production of Nancarrow's player piano works by Sandoval and Höldke [Hoe07]). Where in instrumental music this segregation takes place automatically as each instrument has its own position in space, using loudspeakers, this can be achieved by using separate speakers for separate voices.

1.2 Realisation

The extent to which spatial parameters can play a role in musical composition is dependent on the technology available to achieve spatialisation. In the following an overview is given of common technologies and their advantages and disadvantages.

1.2.1 Stereophonic techniques

In the development of loudspeaker technology, there has been a development from mono, to stereo, and to multichannel stereophony (quadraphony, octaphony, 5.1 surround, 7.1 surround, etc.). In its simplest form stereophony is based on amplitude based panning, and in a more complex form on vector based panning, where time delays of the signals are also taken into account.

In electro-acoustic music, we can see that in the development of using these techniques there has been a great interest in how to control the panning of sound sources, in a way that goes beyond the common "pan-pot" or balance knob on a mixing desk. Jacques Poullin (1951) created in Paris the *potentiomètre d'espace* system for distribution of sound amongst four loudspeakers, two in front of the audience, one above and one in the rear. This system allowed the performer to position a sound by simply moving a small hand-held transmitter coil toward or away from four large receiver coils that were positioned around him. The experiments with spatial projection of sound induced conceptual ideas on the spatial parameter of electronic music, that went beyond a notion of stereophony,

Figure 1.1: The inside of the Osaka spherical pavilion. The speakers are placed on the sphere, in a similar pattern as the lights.

as Pierre Schaeffer (1951) [Sch90] mentions: "... le nouveau procédé est une dialectique du sons dans l'espace et je pense que le terme de *musique spatiale* lui conviendrait mieux que celui de stéréphonie."[4].

Though most commonly quadraphonic or octaphonic setups are found, there are examples of works that used many more speakers, such as the multimedial work *Poème Electronique* (1958) of Edgard Varèse, hundreds of loudspeakers were used to create spatial trajectories through the Philips Pavilion (by LeCorbusier and Iannis Xenakis), and the hemisphere spatialisation for the World Expo in Osaka (1970). The latter was a further development of Stockhausen's technique for *Kontakte* (1958-60), where he rotated one, highly directional, speaker on a plate, playing sound, and recorded this with four microphones onto four channels [MB88]. At the start of 1960 Manfred Krause implemented an improved version of this idea on a electronic basis (the *Tonmühle*): the electronic signals are divided over several speakers by changing a resistance. The number of speakers was increased, and the device was usable for live performance [GGF96]. The basics for this device were developed further for the hemisphere spatialisation used for the World Expo in Osaka in 1970 (figure 1.1). 50 loudspeaker groups were divided over a hemisphere and a corresponding ball with 50 buttons to control the spatialisation was developed [GGF96] (figure 1.2).

Recently, there is a renewed interest in the hemisphere for spatialisation,

[4] "...the new process is a dialectic of sound in space and I think that the term *spatial music* could fit it better than stereophony"

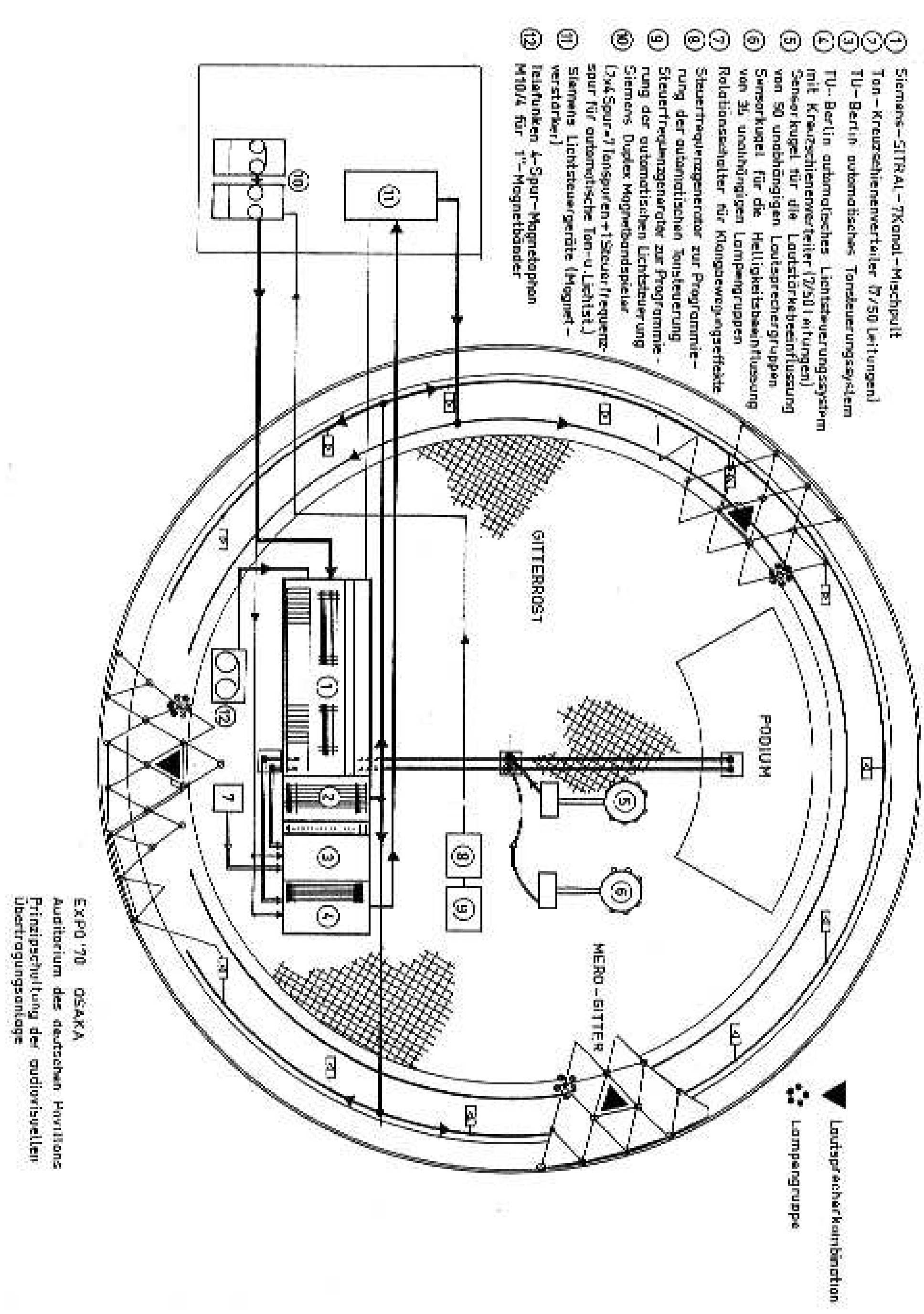


Figure 1.2: Schematic overview of the audio-visual technology for the sphere-pavilion in Osaka.

like for example the *Klangdom* in the ZKM, Karlsruhe [RGB06]. One of the reasons of this development could be that it is now much easier to control the spatialisation over such large systems, as the signals can be calculated by software on a commercially available computer (see for example [Bar07]), and are no longer dependent on specially designed hardware for the purpose.

Simultaneously, the surround formats introduced by the film industry, such as 5.1 surround and 7.1 surround, are gaining some popularity in the field of electro-acoustic music, as it provides a means for a large scale distribution of the works, made available through the DVD format, as well as an interest of radio stations to broadcast in these surround formats. Nonetheless these surround formats do pose some problems, as they were primarily developed for a frontal presentation, and as a consequence, there are more speakers located in the front of the listener (where the visual focus is), than in the rear, and moreover in the compressed format the rear channels are treated as less important and thus more compressed during the encoding. Thus although the technology provides a means for distribution, the composer has to take into account that he has to compose for an unbalanced space, where the front is more important than the rear. It is hard, if not impossible, to place sounds at the sides, as the angles between the front-left and rear-left (and also front-right and rear-right) are too large for stereophonic panning.

The main problem for using quadraphony [Kop05, e.g.] was not only that manufacturers did not agree upon a common format for distribution, but also that the angle between the four speakers (one in each corner of the hall) are actually too wide to achieve a good audible result: between all speaker pairs, there is a "hole" when the source is in the middle between the speakers. Moreover, the actual intended result is only heard properly in the centre of the four speakers, the so-called *sweet spot*. However in many concert situations for electro-acoustic music, this spot is reserved for the composer or interpreter controlling the mixing desk, and no audience member is actually seated in the sweet spot [B93̈]. Octaphonic setups (8 speakers equally spaced in a circle around the audience) remedy the situation only partially, as the angles between the speakers are made smaller, the "hole" in the middle between the speakers is not as serious. Still there is a large dependency on the sweet spot.

1.2.2 "Put a loudspeaker there"

The weaknesses of stereophonic techniques have prohibited some composers (for example G.M. König) from using these techniques for spatialisation, as they do not feel comfortable with the result. Curtis Roads mentions in a personal conversation with Alberto de Campo:[5] "If you want the sound to come from a specific place, put a loudspeaker there".

Examples of these can be found in compositions or sound installations created for specific spaces, where speakers are placed at specific locations within the space, and are used to emphasise or use the natural acoustics of the space,

[5]ca. 2000, probably at Sound in Space Conference, Santa Barbara. (source: email communication between the author and De Campo).

by infusing acoustic energy into the room at those places where it will resonate well with the acoustics.

1.2.3 Sound diffusion

A different approach, more as a way of performing an electro-acoustic piece, than seeing the spatialisation as part of the composition, diffusion techniques have been developed using so-called loudspeaker orchestras, like the Birmingham Electro-Acoustic Sound Theatre (BEAST[6]), the Groupe de Musique Expérimentale de Bourges' GMEBaphone [Clo01] and of the GRM[7].

While the main technique used to disperse the music over the speakers, is essentially amplitude panning, the loudspeakers are chosen for their special characteristics, i.e. not all speakers are the same. Each of the speakers is seen as a different interpreter for the music and both the spatial dislocation of the speakers, as well as their individual spectral and directional characteristics, play an important role in this tradition.

1.2.4 Doppler effects

Chowning (1971) [Cho71] described methods to create the impression of fast moving sound by simulating Doppler effects.

1.2.5 Delay and reverberation

One way of adding spatial features to a sound is to mimic the acoustic reverberation of a hall. Technologies [WT06] to achieve this have been developed for a long time, starting with rather physical solutions, where the sound was broadcast into a reverberation room and recorded again, as well as inputting the sound into a metal plate (e.g. the EMT Hallplatte from 1957), and recording it again from another location on the plate. Later analogue electronic solutions were developed, or mixed analogue-digital solutions, through devices like bucket brigade delays. Echoes can also be created by sampling and playing back the sounds. As digital signal processing developed itself further, DSP processors with various reverberation algorithms became available on the market, and in recent years convolution has come within the realm of realtime processing, increasing the popularity of the convolution reverb.

1.2.6 Decorrelation

Decorrelation of audio signals [Ken95, Vag01] creates timbral colouration and combing due to constructive and destructive interference, produces a diffuse sound field and does not suffer from image shift and the precedence effect. This technique does not aim to create a physically correct image, but rather uses psycho-acoustic effects (confusion) to achieve a diffuse, broad sound image.

[6] `http://www.ea-studios.bham.ac.uk/BEAST/`
[7] `http://www.ina.fr/grm/`

1.2.7 Metaphoric techniques

The spatialisation technique does not necessarily have to be related to natural spatial sound, as for example shown in the software Meloncillo[8] which determines the strength of the certain sounds in certain speakers based on an arbitrarily defined function around the speaker [Rut04].

Here space is used in a more abstract sense, as a way of navigating through parameters, in this case the amplitude of sounds for a certain loudspeaker. A more generalised approach for using spatial relationships for navigating through parameter spaces is found in [MW03], though this goes beyond the scope of this thesis.

1.2.8 Ambisonics

The ambisonic sound system [MM95] is a two-part technological solution to the problem of encoding sound directions and amplitudes, and reproducing them over practical loudspeaker systems, so that listeners can perceive sounds located in a three dimensional space. This can occur over a 360 degree horizontal only sound stage (pantophonic system), or over the full sphere (periphonic system). The system encodes the signals in a so-called *B-format*; the first order version of this encodes the signal in three channels for pantophonic systems and a further channel for the periphonic, i.e. "with-height" reproduction.

Essentially the system gives an approximation of a wave field by a plane wave decomposition of the sound field at the listener's position. This approximation gets more precise, when the order of ambisonics is increased, which also means that more channels are needed for encoding, and that more speakers are needed for decoding. For very high orders, ambisonics can be equated to wave field synthesis [DNM03].

In recent years, ambisonic techniques have become more and more popular, as a way of spatialising sound. Most commonly first order systems are found, in rarer cases higher order ambisonics. With ambisonic techniques different spatialisation effects can be achieved than with stereophonic techniques, though, as stereophony, it is dependent on a "sweet spot" where the effect is optimal.

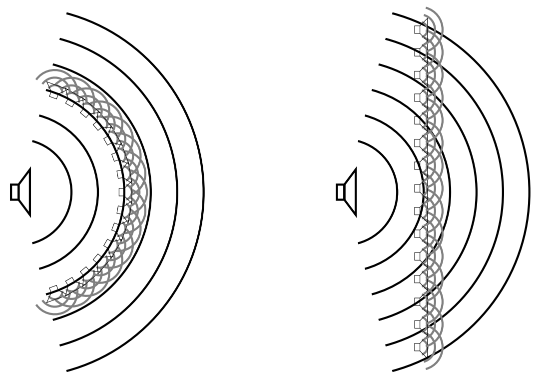
Implementations of ambisonics can be found in many popular sound synthesis programs, such as CSound, Max/MSP, Pd and SuperCollider[9].

1.2.9 Wave Field Synthesis

Wave Field Synthesis (WFS) was introduced by Berkhout [Ber88] in 1988, and is an approach to spatial sound field reproduction based on the Huygens principle. With WFS it is possible to create a physical reproduction of a wave field; this has the advantage over other techniques in that there is no *sweet spot*, i.e. there is no point in the listening area where the reproduction is remarkably better than at other places, rather, there is a *sweet area*, that is quite large.

[8] `http://meloncillo.sourceforge.net`

[9] `http://www.csound.org`, `http://www.cycling74.com`, `http://www.puredata.org` and `http://supercollider.sourceforge.net` respectively.



(a) The Huygens' Principle (b) Wave Field Synthesis

Figure 1.3: From the Huygens' Principle to Wave Field Synthesis

The principle of Huygens states that when you have a wave front, you can synthesise the next wave front by imagining on the wave front an infinite number of small sound sources, whose waves together will form the next wave front [Huy90] (figure 1.3a). A listener will then not be able to determine the difference between a situation where the wave front is real, or when it is synthesised.

This principle can be translated to mathematical formulae using theories of Kirchhoff and Rayleigh and can then be applied for use with an linear array of loudspeakers (as described in chapter 2). By sending the right signals to each loudspeaker, that is: sending to the loudspeaker the signal that would be measured at the location of the speaker, if the sound source would really be at the location it is virtually placed, sound sources can be virtually placed at any place in a horizontal plane behind, or even in front of, the speakers (see figure 1.3b).

In comparison with ambisonic techniques, wave field synthesis is better at reproducing spatial depth, though the vertical dimension which can be used in ambisonics is something that is lacking in WFS.

The use of Wave Field Synthesis for electro-acoustic music is one of the main topics of this thesis and thus will be elaborated upon in the next chapters.

1.2.10 Focus of software tools

Several aspects of spatialisation are found in compositions and software, creating

- spatial trajectories of sound sources,
- an acoustic environment of sound sources, by reflections and reverb,
- a broad sound image,
- a diffuse enveloping sound field.

Perceptual parameters	azimuth and elevation source presencedistance brilliancewarmth room presence running and late reverberation envelopment heaviness and liveness
Low-level DSP parameters	equalisation Doppler effect air absorption multi-channel reverberation directional distribution according to various setups (stereo, 3/2 stereo, multi-channel system (4 to 8 speakers), 3-D stereo either binaural using headphones or transaural using speakers)

Table 1.1: The parameters that can be set in IRCAM's *spat*. (*Source:* `http://forumnet.ircam.fr/356.html?&L=1`, accessed at August 10, 2007)

Most traditional audio editors are modelled after the work flow of working with tape recorders and mixing desks, you will find the audio tracks on the one hand, and mixing controls on the other hand, with possibilities for automation of the mixing controls. Usually the panning of a sound source can be automated, which for a two-channel stereo pair is still relatively easy to visualise. However for the representation of spatial trajectories for more than two channels, one spatial coordinate is not sufficient. For movement in a horizontal plane, at least two dimensions are needed, which in a spatio-temporal visualisation results in three dimensions, which can be visualised using 3D graphic techniques (see chapter 3), but poses challenges to the interaction (editing of) with this representation. An alternative approach (as found in Meloncillo [Rut04]) is to provide several representations: one in a traditional 2D time-spatial coordinate view, and a 2D spatial view of a selected portion of the time line. In Snoei's software (see chapter 4) for Wave Field Synthesis, extensive work has been done creating compositional tools, where paths can be edited in 2D views, but also speed curves of movement along the paths can be edited.

Reverb tools are commonly found as plugins, for one channel, or two channel use. A multichannel example for adding reverberation to an audio channel, can be found in the *spatialisateur* (or *spat*), created by IRCAM; in this tool, parameters can be set for various perceptual aspects of the acoustic environment, such as presence, envelopment and so on (for a complete list see table 1.1). The result can be rendered for several loudspeaker setups.

Currently, there is a development from the traditional track/channel oriented way of working, where an audio track corresponds to what is played on a certain speaker channel, to an object oriented way of working, where the audio track is treated as a sound source object, and the spatial characteristics that are needed for the reproduction of the source are stored as meta data, so that the reproduction method is independent of the encoding (e.g. MPEG4 [PBG03], and XML3DAudio [Pot06]). This has as an advantage that spatial compositions are interchangeable between systems, as well as a reduction of stored audio channels for systems that use a lot of loudspeakers.

1.2.11 Summary

In most of these realisation techniques, the loudspeaker is seen as a neutral instrument, with which a simulation or illusion can be achieved. In the discussion on sound diffusion, we saw that loudspeakers were chosen for their different characteristics as well, which leads to the next part of the introduction, where the use of alternative transducers and the need for control over directional behaviour of the sound source is discussed.

1.3 Sound sources

1.3.1 Alternative transducers in sound art

Since the 17th century there has been a development of automated music machines, which in the late 19th century had its peak, as automated orchestra automata were developed, to enable cafés to have entertainment music, without having to hire musicians. The commercial development of these machines stopped more or less, as the gramophone proved to be a much cheaper solution[10].

Nevertheless, some artists, such as Godfried Willem Raes[11], Jacques Rémus[12] and Trimpin[13], have since continued creating such machines, for various reasons. One of the reasons is that they do not like the generic sound of a loudspeaker and believe that a much richer sound can be achieved by the use of other electro-mechanical means[14]. By exciting objects of various materials in one way or another different types of sound can be created, which have spatial characteristics both by the dimensions of the objects themselves, as well as the placement of several of these objects spread out in a space, which creates a spatial sonic environment. Another point of interest, but which is not so relevant for our current discussion, is that the intention is to create machines which play acoustical instruments in ways humans are not capable, due to the limitations of the human body.

The *intonarumori* as developed by Luigi Russolo and Ugo Piatti in 1913 [AB01] can also be seen in this context: they attempted to create instruments which created different sounds than conventional acoustical instruments.

Paul de Marinis[15] used gas flames in his piece *Firebirds* (2004), where the

[10] Source: guided tour of the museum "Van Speelklok tot Pierement" in Utrecht, the Netherlands.

[11] `www.logosfoundation.org`

[12] `http://www.mecamusique.com`

[13] `http://www.otherminds.org/shtml/Trimpin.shtml`, `http://en.wikipedia.org/wiki/Trimpin`

[14] source: the NIME 2007 workshop and panel on musical robotics.

[15] `http://www.well.com/~demarini/exhibitions.htm`

gas flames are "suitably modulated by electrical fields to be made to act as omnidirectional loudspeakers of surprising clarity and amplitude". In his piece *Raindance* (1998), he used water valves which were opened and closed, to create rhythmic versions of popular rain songs, which could be heard by the visitors, as they walked underneath the valves with an umbrella.

Edwin van der Heide[16] uses air pressure valves, whose opening and closing rate can be modulated, in his piece *Pneumatic Sound Fields* (2006). Here also the valves are spread out over a larger space, in order to create a spatial composition.

Julius Stahl's *Ensemble readymade*[17], in which exciters are placed on objects found at the performance site, which will then act as sound sources. In his piece *Tocar* (2006), he uses exciters placed on 12 platforms spread out in a matrix form in the room in which the concert or installation takes place, and the audience is seated on these platforms. The sound material, played by a percussionist and pre-recorded sounds, is spatialised in various forms over the matrix. The audience sitting on the platforms, will both hear the sound transmitted this way into the room, and feel the vibration of the platform on which he or she is sitting, thus creating an individual audio-haptic experience for each member of the audience. This can be seen as a further excursion in the direction in which Xenakis was working (see above), by not only introducing a more individual auditory perspective, but also a haptic experience.

Savannah Agger uses plates of various metals in her piece *Metal Degrees* (2007), as physical sound processors[18], by infusing the plates with sounds through speaker cones, and recording the results again with microphones, and amplifying these through normal loudspeakers. Here the plates are not used so much as sounding objects, but rather as processing units, which due to their physical nature, have a certain unpredictability, due to factors that influence the metal's behaviour, such as temperature, which cannot be controlled by the composer.

1.3.2 Source directivity

A common problem that has been noticed by composers and listeners, is that the sound from acoustical instruments and those from electro-acoustic parts of a composition have a very different quality; often the sound of the acoustical instrument sounds much more rich and lively, than the sound from the loudspeaker. This is a result of the directivity of the instrument, which interacts with the acoustics of the room and which is often (consciously) varied while playing as the performer moves his instrument, and (in contrast) the static directivity of the loudspeaker: the loudspeaker itself is static, and there is no variation in directivity as a result of the kind of sound that is played, while different acoustical instruments (producing different timbres) also have different

[16] `http://www.evdh.net`

[17] `http://www.juliusstahl.de/ensemble.html`

[18] comparable with the original plate reverbs, though Aggers aim is not to create a realistic sound reverb, but rather she is interested in the different characteristics of the different materials.

Figure 1.4: The 120 independent element spherical loudspeaker array developed at Berkeley's CNMAT. (*Photo from CNMAT*, `http://www.cnmat.berkeley.edu/Research/AES2006/speakerarray/icosaspeakers.htm`)

directivities.

A sound source has a certain directivity, i.e. when an object emits sound into space, it is generally not emitted equally strong in all directions. In the far field[19] of the sound source, the source can be described as a point source with a directivity function, usually a function of angle and frequency.

The directivity of various acoustical instruments has been studied in depth by Meyer [Mey95] and Fletcher [FR98], analysing the (temporal) directional characteristics of musical instruments. Misdariis [MWC01] and Warusfel [WM01, WCMC04] have proposed special 3D loudspeaker arrays to simulate and control different radiation characteristics for the purpose of electronic music. Freed et al. [FAWK06] created a spherical speaker array consisting of 120 speakers with programmable radiation control for the same purpose (figure 1.4). Commercially available is a 6 channel half-sphere loudspeaker from the company Electrotap[20].

[19] the far field is at a distance that is much larger than the dimensions of the object and/or much larger than the wavelengths of the sound produced by the object.

[20] `http://www.electrotap.com/`

(a) Coherent sources (b) Incoherent sources

Figure 1.5: Influence of coherence on perceived source extent (indicated with the dotted line).

1.3.3 Source size and shape

The source directivity function is defined as an approximation of the source radiation in the far field of a source. When getting closer to the source, the actual shape and size of the object play an important role in its perception, as we will no longer get the impression of a sound coming from one place, but rather from a wider region.

Source extent has been studied in the literature under the names of apparent source width, tonal volume and others (see e.g. [Bla97] for an overview). It has been shown that the perceived source extent depends on the value of the interaural cross correlation coefficient (IACC) [Mor02], sound loudness [Bor26], pitch and signal duration [PB82].

In order to reproduce the extent of a sound source, several, spatially distinct, point sources can be used; the sound from these point sources must be statistically uncorrelated from one another, while as the correlation is high between the point sources, they will be perceived as a single auditory event [Bla97], localised at the centre of gravity (dependent on the positions of the point sources and their intensity) as indicated in figure 1.5. If the signals are weakly correlated they will be perceived as distinct auditory streams and a spatially wide sound source will be perceived. If the point sources are densely distributed, not every single point source will be perceived as a different auditory stream, but an impression of a single, spatially large, sound source will be produced.

Potard [Pot06] did some further investigations into the perception of source size and shape, which will be reviewed and criticised here.

Horizontal extent

His first experiment studies the perception of horizontal source extent. He found that with increasing density of the point sources the source width was more underestimated. He ascribes this to a higher IACC[21] value, when more sources are present. On the other hand, when the point sources are further apart with increasing density, with a source width above 100 degrees (reproduced by 3 angular equidistant sources) a gap is perceived and the sources are distinguished

[21] Interaural Crosscorrelation Coefficient, for a definition see [And85]

individually. Another important factor he found, was the signal type, which may be linked to the tonal volume of the source.

Here should be remarked that for the test setup he used (7 speakers in an arc, spaced 30 degrees from each other), the stimuli with a density of one source per 10 degrees, could only be reproduced by relying on stereophonic panning. Wegmann [Weg05, ch. 4] showed that the directional perception of sound based on stereophonic techniques is probably based on a quite different principle, than the directional perception of a real sound source, though it is not yet understood exactly this works. It is therefore questionable whether the mix of approaches in Potard's experiment is valid.

Horizontal and vertical extent

In a second experiment both the perception of horizontal and vertical extent is studied, using a 16-speaker array placed in a geodesic dome configuration (CHESS). He found that the perceived horizontal extent of the sources matched coarsely with the intended extent. A perceptual narrower extent was noticed with sources with a horizontal extent of 60 degrees. He ascribes this to the high density of the point sources (15 degrees), but again it should be remarked that the minimum angle between loudspeakers of the test setup was ca. 30 degrees, so reproduction of stimuli with a higher density than this were relying on some kind of panning to be achieved.

The perception of one-dimensional vertical sound sources matched the intended vertical extent fairly well, if one takes into account that localisation in the median plane is less accurate [Bla97]. He also found that there is a notable difference between the accuracy of source extent perception on the side and behind the listener: the perception is less accurate, just as with localisation of sound sources.

For sources with a rectangular 2D extent the perceived and intended extent matched well. Generally the subjects could distinguish well between 2D sound sources, horizontal or vertical line sources and point sources.

Shape

His third experiment focused on the ability to identify source shapes. The speaker setup is indicated in figure 1.6. On this setup six different source shapes were presented between which the listeners had to distinguish, for four different types of signals: white noise, lowpass filtered noise, highpass filtered noise and a blues guitar riff. The source shapes were created by sending decorrelated signals to one or more speakers, in such a way that there was no difference in loudness between the stimuli.

The point source (sound coming from one speaker) was clearly identified by the subjects. For white noise and high pass noise presented to the front of the listener, the identification of the sound source shape was well above statistical probability, but still under the 50 % threshold. The listeners could identify well between different types of shapes: point source, line source and 2D sources. Source shape identification was also better for sources presented

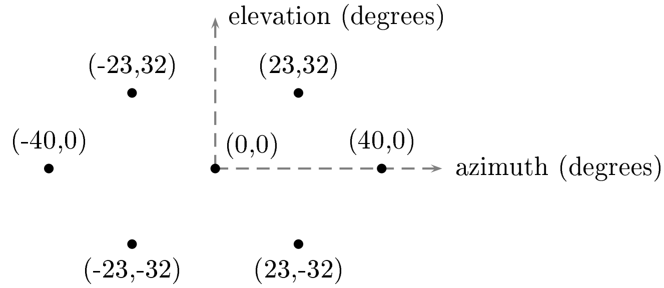


Figure 1.6: Speaker coordinates for the shape experiment by Potard [PB03]. The speakers were at 1.6m from the listener position.

in the front than in the back; in fact, source shapes presented at the back of the subject were only identified significantly correct for the point source. For frontal presentation only the source shape, where all speakers were used, was not identified with statistical significance.

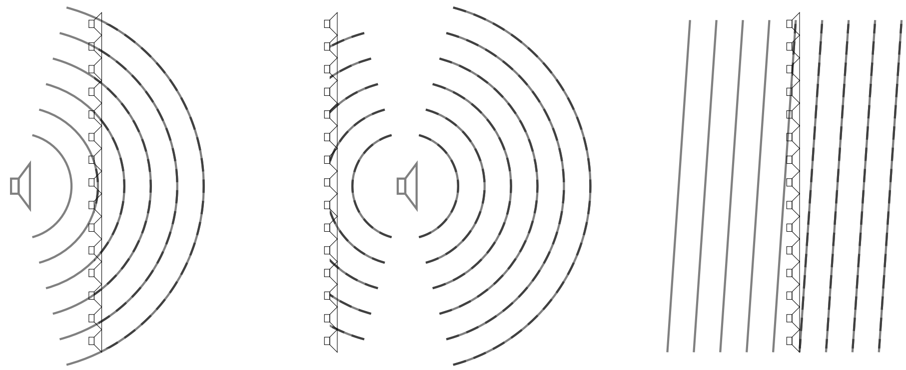
In his discussion to explain the better shape identification for white noise and highpass filtered noise, than with the other signals, he notes that with the method used to create the source shapes, the method of decorrelation, the phases of the signals get randomised, causing the binaural system to have to rely on interaural level difference (ILD) rather than interaural time difference (ITD). ITD cues are known to be mainly used for localisation of sounds with frequencies below 3 kHz [Bla97].

Naturalness of extended sources

In a further experiment Potard studied whether 3D audio scenes using extended sources were perceived to be more natural than those only using point sources, and found that they were perceived to be more natural in 70.4 % of the time.

Summary of experiments on source size and shape

From this overview, we can motivate the further investigation of reproducing the spatial extent of sound sources, as the spatial extent is apparently perceptually relevant, and can lead to an increase in perceived naturalness of an audio scene. Potard's studies have focused on the use of decorrelated point sources to create spatial extent. He already noted that this may lead to loss of phase information which may carry important cues for perception of spatial extent for frequencies below 3 kHz. It should also be noted that decorrelated sound sources correspond to sources with spatial extent, which are actually built up from large amounts of small sources, for example rustling of leaves of a tree, a murder of crows, an applauding crowd or a singing choir, that are physically uncorrelated: they make a similar sound, but there is no structure borne physical relation between the sounds (see section 3.11). For a sounding object, which is made from connected materials, the sound from different points on the object will be correlated and



(a) Point source behind the loudspeakers

(b) Point source in front of the loudspeakers

(c) Plane wave

Figure 1.7: Source types that are currently implemented in many Wave Field Synthesis applications

have a physical relation with each other.

1.3.4 Sound source occlusion

In natural environments sound sources can be occluded by other objects, which means that the sound will be diffracted by the occluding object and only a part of the sound will reach the listener. An overview of diffraction models is given in chapter 5. The perception of occluded sources was investigated by Farag et al. [FBA03], who did listening tests on the (horizontal) localisation of sounds that are occluded by objects. The results of their experiments show that the perceived location tends to be towards the edge of the occluding object, that is closed to the sound source. They attribute this to the precedence effect. For a source at equal distance from both sides, the listener localises the source more or less in the middle, as a summation effect takes place, as the sound from both edges arrives at the same time.

1.3.5 Source characteristics in WFS

Currently, in many implementations of WFS only point sources and plane waves have been implemented as source types (see figure 1.7).

Theile [TWR02] introduced the Virtual Panning Spots (VPS) as a method to create broad sound images: you create two point sources, put them on a specific location and pan the sound between these two points. This is a relative simple solution to create a broad sound image, which has the advantage that it can make use of traditional sound recording techniques and studio techniques, i.e. traditional audio engineers are familiar with the technique and can use their accumulated experience.

Bork [BK03] did an experiment to reproduce the source radiation of a grand piano by using a number of point sources, placed so that the resultant radiation characteristic corresponds to that of a real grand piano.

Corteel [Cor07] (IRCAM) describes how directional source characteristics (as a combination of multipoles) can be reproduced and used both for creating direct sound with directive characteristics [WCMC04], as well as a tool to influence the reverberant field in the listening room [CCW03, WM04], by using a directive source to emit more energy to the walls of the room, and thus creating more lateral energy in the listening area. In order to optimise the amount of energy that is emitted by the speakers to the listening area, multichannel equalisation techniques are used.

Chapter 2

Theory

2.1 Homogeneous Wave Equation

The acoustic wave propagation in air can be described by the homogeneous acoustic wave equation

$$\nabla^2 p(\vec{x}, t) - \frac{1}{c^2} \frac{\partial^2}{\partial t^2} p(\vec{x}, t) = 0 \tag{2.1}$$

where $p(\vec{x}, t)$ is the acoustic wave pressure at the position $\vec{x}$ and time t, c denotes the speed of sound, and ∇^2 is also referred to as the *Laplace* operator $\triangle = \nabla^2$. This equation describes the wave nature of sound, including diffraction, provided the following assumptions are valid:

1. the propagation medium is homogeneous,
2. the propagation medium is quiescent,
3. the propagation medium can be characterised as an ideal gas,
4. the state changes in the gas can be modelled as adiabatic processes,
5. the pressure and density perturbations due to wave propagation are small compared to the static pressure p_0 and the static density ρ_0.

An alternative form of equation (2.1) can be obtained when a Fourier transformation is made with respect to the time t of the acoustic pressure $p(\vec{x}, t)$. The Fourier transformation pair of $p(\vec{x}, t)$ is given by:

$$P(\vec{x}, \omega) = \mathcal{F}_t\{p(\vec{x}, t)\} = \int_{-\infty}^{\infty} p(\vec{x}, t) e^{-j\omega t} dt \tag{2.2}$$

$$p(\vec{x}, t) = \mathcal{F}_t^{-1}\{P(\vec{x}, \omega)\} = \frac{1}{2\pi} \int_{-\infty}^{\infty} P(\vec{x}, \omega) e^{j\omega t} dt \tag{2.3}$$

where $\omega = 2\pi f$ denotes the temporal (radial) frequency and $\mathcal{F}_t\{\cdot\}$ the Fourier transformation with respect to the time t. With this transformation the acoustic wave equation can be formulated in the frequency domain

$$\nabla^2 P(\vec{x}, \omega) + k^2 P(\vec{x}, \omega) = 0 \tag{2.4}$$

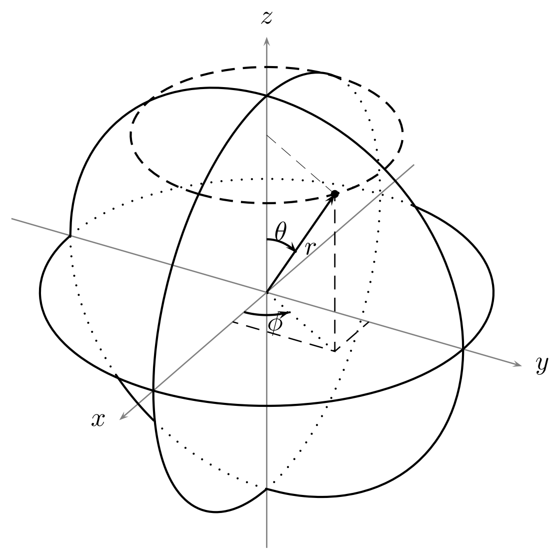


Figure 2.1: Spherical coordinate system. The coordinates of a point with cartesian coordinates (x, y, z) are given by $r = \sqrt{x^2 + y^2 + z^2}$, $\phi = \mathrm{atan}(y/x)$ and $\theta = \mathrm{acos}(z/r)$. The inverse equations are $x = r\cos\phi\sin\theta$, $y = r\sin\phi\sin\theta$ and $z = r\cos\theta$.

where $k = \frac{\omega}{c}$, the acoustic wave number.

2.2 Inhomogeneous Wave Equation

If there is a source present in the medium, equation (2.1) has to be adapted to reflect the addition of energy to the medium by the source:

$$\nabla^2 p(\vec{x}, t) - \frac{1}{c^2} \frac{\partial^2}{\partial t^2} p(\vec{x}, t) = -q(\vec{x}, t) \tag{2.5}$$

In the following, solutions to this equation will be discussed.

Point source

Consider a radially oscillating sphere, that generates an outgoing wave. A point source is a model of the limiting case as the radius of the sphere becomes smaller and smaller. The sphere will degenerate to a single point in space. Then $q(\vec{x}, t)$ becomes a *Dirac impulse* at position $\vec{x}$ and time t

$$\nabla^2 p(\vec{x}, t) - \frac{1}{c^2} \frac{\partial^2}{\partial t^2} p(\vec{x}, t) = -\delta(\vec{x}, t) \tag{2.6}$$

It is convenient to use a spherical coordinate system (see figure 2.1) to describe the field caused by a point source, due to the omnidirectional nature of the radiated pressure field. The acoustic pressure field P_p for a source in the origin can be described by

$$P_p(\vec{x},\omega) = P_p(\rho,\omega) = \hat{P}(\omega)\frac{1}{\rho}e^{-jk\rho} \tag{2.7}$$

where k denotes the wave number, ρ the radius and $\hat{P}(\omega)$ a frequency dependent pressure amplitude. Transforming equation (2.7) back into the time domain using the inverse Fourier transformation (2.3) we obtain

$$p_p(\vec{x},t) = p_p(\rho,t) = \frac{1}{2\pi\rho}\hat{p}(t-\frac{\rho}{c}) \tag{2.8}$$

Arbitrary source

An arbitrary shaped source can be approached by a point source if the dimensions of the source are small compared to the considered wavelength and the wave field is observed at a large distance compared to the source dimensions. However, in most cases the field will not be spherically symmetric as the source may not emit the same sound pressure in each direction. Thus we have to add a dependence on direction to equation (2.7)

$$P_p(\vec{x},\phi,\theta,\omega) = P_p(\rho,\phi,\theta,\omega) = \hat{P}(\omega,\phi,\theta)\frac{1}{\rho}e^{-jk\rho} \tag{2.9}$$

$\hat{P}(\omega,\phi,\theta)$ can be split into a directivity function $G(\omega,\phi,\theta)$ and an amplitude function $P_a(\omega)$ as follows

$$\hat{P}(\omega,\phi,\theta) = G(\omega,\phi,\theta)\hat{P}_a(\omega) \tag{2.10}$$

If the conditions for the point source approximation are not met, then another approximation for an arbitrary shaped source must be found. As the wave equation is linear, any source can be approximated by a sum of point sources, for example an integral over a surface of infinitely close spaced point sources.

2.3 Kirchoff-Helmholtz integral

Green's second theorem states:

$$\int_V [F\nabla^2 G - G\nabla^2 F]dV = \oint_S [F\nabla G - G\nabla F]\cdot\vec{n}dS \tag{2.11}$$

where F and G must be scalar functions defined inside and on S, with continuous second-order derivatives inside S.

We can choose for F and G the following two functions:

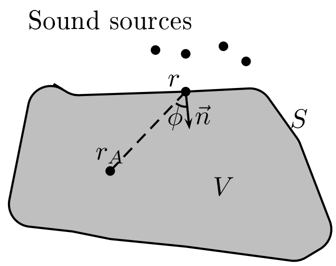


Figure 2.2: The geometry used for Green's second theorem: a volume V surrounded by a surface S

1. F equals the Fourier transformed pressure of a compressional wave field that is generated by sources outside the closed surface S: $F = P = P(\vec{r}, \omega)$. Inside S, P will satisfy equation (2.4).
2. G equals the Fourier transformed pressure of a compressional wave field that is generated by a (virtual) monopole in a point A inside S:

$$G = \frac{e^{-jk\Delta r}}{\Delta r} \tag{2.12}$$

with $\Delta r = \sqrt{(x_A - x)^2 + (y_A - y)^2 + (z_A - z)^2}$. G satisfies

$$\nabla^2 G + k^2 G = -4\pi\delta(\vec{r} - \vec{r}_A) \tag{2.13}$$

Substituting these expressions for F and G in Green's theorem (2.11) yields

$$\begin{aligned} -4\pi \int_V P(\vec{r}, \omega)\delta(\vec{r} - \vec{r}_A)dV &= \oint_S [P\nabla G - G\nabla P] \cdot \vec{n} dS \\ -4\pi P(\vec{r}_A, \omega) &= \oint_S [P\nabla G - G\nabla P] \cdot \vec{n} dS \\ P(\vec{r}_A, \omega) &= \frac{1}{4\pi} \oint_S [P\frac{\partial G}{\partial n} - G\frac{\partial P}{\partial n}]dS \qquad (2.14) \end{aligned}$$

using the equation of motion

$$-\frac{\partial P}{\partial n} = \rho_0 \frac{\partial v_n}{\partial t} \tag{2.15}$$

$$\begin{aligned} P(\vec{r}_A, \omega) &= \frac{j\omega\rho}{4\pi} \oint_S [PG_n - GV_n]dS \\ &= \frac{1}{4\pi} \oint_S [j\omega\rho_0 V_n(\vec{r}, \omega) + P(\vec{r}, \omega)\frac{1 + jk\Delta r}{\Delta r} cos\phi] \frac{e^{-jk\Delta r}}{\Delta r} dS \end{aligned} \tag{2.16}$$

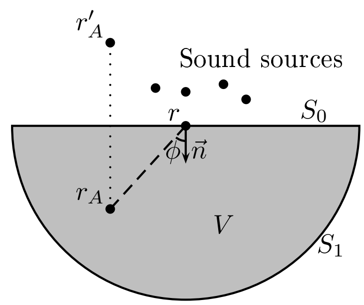


Figure 2.3: The geometry used for deriving the Rayleigh integral: a volume V enclosed by a a plane surface S_0 and a hemisphere S_1. r'_A is the mirror point of r_A in the surface S_0.

where $G_n = \partial_n G / j\omega\rho$ is the normal component of the monopole particle velocity on S.

Equation (2.16) is the 3D forward Kirchhoff-Helmholtz integral. It states that in a volume V of a source-free compressional, homogeneous and isotropic medium the sound pressure at any point can be calculated if both the pressure and normal velocity on the boundary surface S is known.

2.4 Rayleigh integrals

In order to simplify the Kirchhoff integral a Green's function can be chosen such that it satisfies a more convenient boundary condition. If on S a boundary condition is chosen such that $\partial G/\partial n = 0$, the first term of equation (2.16) vanishes; this implies that S acts as a perfect reflecting rigid surface. If S is a reflection free surface (or *pressure-release*), the boundary condition is $G = 0$ and the second term of equation (2.16) vanishes [Spo05, Sta97].

Consider the geometry of figure 2.3. The volume V is enclosed by a a plane surface S_0 and a hemisphere S_1 with radius r_1. The sound sources are situated in the upper half-space ($z > z_s$), outside V.

Rayleigh I

In the case that $\partial G/\partial n = 0$ on S_0, S_0 is a perfect rigid reflecting surface and a suitable Green's function is the summed pressure field of two equal monopoles situated symmetrically with respect to S_0, i.e.

$$G_{\vec{r}_A}(\vec{r},\omega) = \frac{e^{-jk|\vec{r}-\vec{r}_A|}}{|\vec{r}-\vec{r}_A|} + \frac{e^{-jk|\vec{r}-\vec{r}\,'_A|}}{|\vec{r}-\vec{r}\,'_A|} \tag{2.17}$$

By using this Green's function and letting the radius of S_1 extend to infinity, it can be shown that the integral over S_1 vanishes and we obtain the Rayleigh

I integral:

$$P(\vec{r}_A,\omega) = \frac{j\omega\rho_0}{2\pi}\int_{S_0} V_n(\vec{r},\omega)\frac{e^{-jk|\vec{r}-\vec{r}_A|}}{|\vec{r}-\vec{r}_A|}dS \tag{2.18}$$

this means that any field due to sources in the half space $V'(z < z_S)$ can be reconstructed in the half space $V(z > z_S)$ by means of a continuous distribution of monopoles at the surface S_0 with source strength $jk\rho_0 cV_n(r,\omega)$.

Rayleigh II

Consider the case in which $G = 0$ on the surface S_0. As S_0 is now a perfect reflection free surface, the Green's function is the difference of the pressure fields of two equal monopoles situated symmetrically with respect to S_0, i.e.

$$G_{\vec{r}_A}(\vec{r},\omega) = \frac{e^{-jk|\vec{r}-\vec{r}_A|}}{|\vec{r}-\vec{r}_A|} - \frac{e^{-jk|\vec{r}-\vec{r}'_A|}}{|\vec{r}-\vec{r}'_A|} \tag{2.19}$$

Using this Green's function, and letting the radius r_1 of the sphere S_1 approach to infinity, the Green's theorem reduces to

$$P(\vec{r}_A,\omega) = \frac{1}{s\pi}\int_{S_0} P(\vec{r},\omega)\frac{1+jk|\vec{r}-\vec{r}_A|}{|\vec{r}-\vec{r}_A|}cos\phi\frac{e^{-jk|\vec{r}-\vec{r}_A|}}{|\vec{r}-\vec{r}_A|}dS \tag{2.20}$$

which is called the Rayleigh II integral. It states that any field due to sources in the half space $V'(z < z_S)$ can be reconstructed in the half space $V(z > z_S)$ by means of a continuous distribution of dipoles at the surface S_0 with source strength $P(r,\omega)$.

2.5 Operator for Wave Field Synthesis

In [Ver98] a $2\frac{1}{2}D$-operator was derived for the reproduction of sources within a horizontal plane by a linear array situated in that same plane. This derivation was necessary for a practical implementation of WFS, as using a planar array of speakers is too expensive (both financially and computationally) and in many cases undesirable, as it would (visually) block the primary sources (of importance when WFS is used for amplification).

For WFS reproduction of a 3-dimensional source object this $2\frac{1}{2}D$-operator is not sufficient as in the derivation only sources in the same plane as the array and reference line are taken into account. Thus, in order to derive a proper driver function, we must start anew from the Rayleigh integrals.

The Rayleigh I integral states that if the normal component (pointing towards receiver area) of the velocity, V_n, is known on a plane S (at $z = 0$) caused by a source distribution on the left hand of S, the pressure in point R in the receiver plane can be determined by

$$P(\vec{r}_R) = \frac{1}{2\pi}\int_S j\omega\rho_0 V_n(\vec{r}_S)\frac{e^{-jk\Delta r}}{\Delta r}dS \tag{2.21}$$

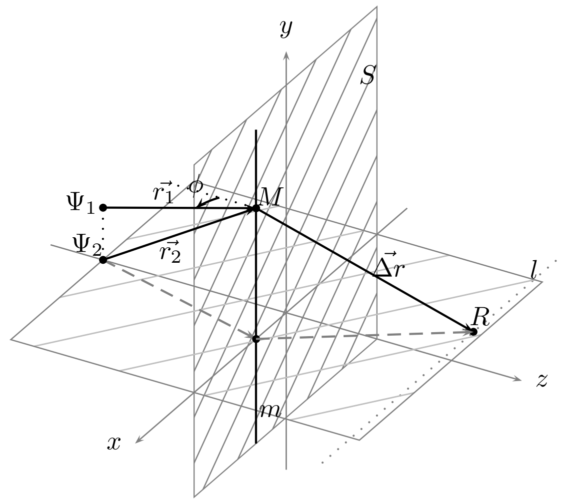


Figure 2.4: Geometry for derivation of the adapted $2\frac{1}{2}$D-operator. Ψ_1 and Ψ_2 are points from the source distribution, $\vec{r_1}$ and $\vec{r_2}$ the vectors to a point M on the integration line m. $\vec{\Delta r}$ is the vector from a point M on the integration line to the receiver point R on the line l.

in which ρ_0 denotes the medium density and $\Delta r = |\vec{\Delta r}| = |\vec{r}_R - \vec{r}_S|$, i.e. the distance from the receiver point $\vec{r}_R = (x_R, y_R, z_R)$ to a point $\vec{r}_S$ on the surface S (see figure 2.4). k is the wave number.

Hence, substituting the far field approximation ($kr \gg 1$) of the normal component of the particle velocity on the plane S due to a monopole sound source into the Rayleigh I synthesis integral yields

$$P(\vec{r}_R) = \frac{1}{2\pi}\int_{-\infty}^{\infty} P_m(\vec{r}_R)dx \tag{2.22}$$

with P_m the pressure on the line m (see figure 2.4):

$$P_m(\vec{r}_R) = \int_m jkS(\omega)cos\phi\frac{e^{-jkr}}{r}\frac{e^{-jk\Delta r}}{\Delta r}dy \tag{2.23}$$

We can now apply the stationary phase method ([Ble84], ch. 2.7) to equation (2.23), as the integrand is of the form:

$$I = \int_{-\infty}^{\infty} f(y)e^{j\psi(y)}dy \tag{2.24}$$

with $\psi(y) = -k(r + \Delta r)$. Now we need to find the stationary point of $\psi(y)$ for which $\psi'(y_0) = 0$. Then, for $|\psi| \gg 1$, the approximate solution to (2.24) is

$$I = f(y_0)e^{j\psi(y_0)}\sqrt{\frac{2\pi j}{\psi''(y_0)}} \tag{2.25}$$

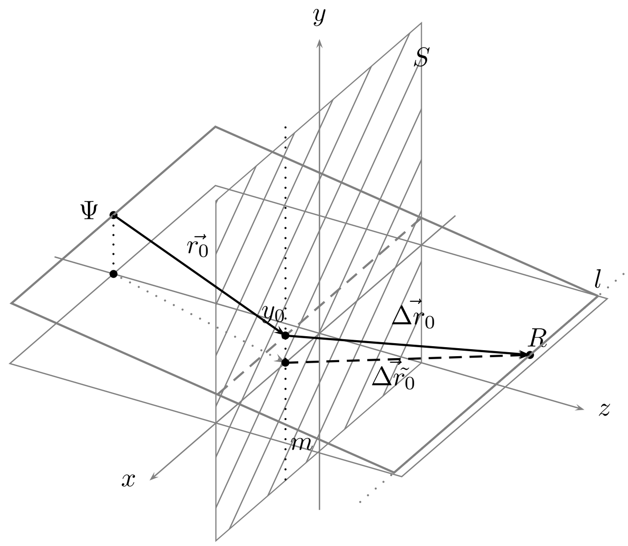


Figure 2.5: The stationary point y_0 lies on the cross-section of plane S and the plane through Ψ and R. In practice $\vec{\Delta r_0}$ will in fact be $\vec{\Delta \tilde{r}_0}$.

where y_0 is the stationary phase point[1] and $\psi''(y_0)$ is the second derivative of ψ evaluated at y_0. $\psi'(y_0) = 0$ is true for the shortest path from the source point Ψ at (x_Ψ, y_Ψ, z_Ψ) to any point R on the receiver line l. The shortest path lies in the plane defined by the source point and the line l, and thus (see figure 2.5):

$$y_0 = y_R + (y_\Psi - y_R)\frac{|z_R|}{|z_\Psi| + |z_R|} \tag{2.26}$$

and at the stationary point:

$$\psi(y_0) = -k(r_0 + \Delta r_0), \tag{2.27}$$

$$\psi''(y_0) = -k\frac{r_0 + \Delta r_0}{r_0 \Delta r_0}, \tag{2.28}$$

$$f(y_0) = \frac{1}{2\pi}\frac{S(\omega)}{r_0 \Delta r_0} cos\phi_0 \tag{2.29}$$

The driver function for a speaker for the contribution of one monopole source point then becomes:

$$Q_\Psi(x, y_0, \omega) = S(\omega)\sqrt{\frac{jk}{2\pi}}\sqrt{\frac{\Delta r_0}{\Delta r_0 + r_0}} cos(\phi_0)\frac{e^{-jkr_0}}{\sqrt{r_0}} \tag{2.30}$$

[1]for each line m ($x = x_m$ and $z = 0$) there is a stationary point with the (vertical) y-coordinate y_0.

where x is the coordinate of the speaker at the speaker array. The speaker should in fact be at the position of the stationary point, $(x, y_0, 0)$, but is in practice placed on the x-axis (at $(x, 0, 0)$). Due to this, several errors are introduced, which will be discussed in the next section.

It should be noted that the actual elevation will not be heard by the receiver, when the elevated source is played back by the WFS-array. However, for the source model which will be presented in chapter 5, the elevated points are of interest, because their contributions will interfere with those of the points in the horizontal plane.

Source points in front of the array

For source points in front of the array, we find in analogy to the derivation of the focused operator in [Jan97]:

$$y_{0,\text{focused}} = y_\Psi + (y_\Psi - y_R)\frac{|z_\Psi|}{|z_R| - |z_\Psi|} \tag{2.31}$$

which means that the stationary point will have a higher elevation than the elevation of the actual source point, as it lies in the plane that goes through the reference line and the source point.

The vertical dimension

The vertical dimension has up to now not really been implemented for WFS, due to several practical considerations:

- In theory one needs a plane of loudspeakers to do this, which has a problem that even more hardware and processing power is needed.
- A plane of speakers renders itself impractical for combination with theatre and/or film, as the stage or screen would be blocked by the speakers.
- The human hearing is less sensitive to vertical positioning of sound sources. Moreover, the perception of vertical position is based on spectral filtering of the sound by the ear pinna[1]. Due to spatial aliasing with WFS, this may be more difficult to achieve.

Currently, research is being done at the TU Delft on adding at least a little bit of vertical information to the wave field, by adding loudspeaker arrays at the ceiling. This way, ceiling reflections may be added to the wave field.

The adapted driver function (2.30) could be used for a number of linear arrays that are distributed at several heights. The location of the stationary phase point could then be a measure for a panning between two vertically adjacent

[1] as this filtering is different for each individual it is unlikely that a sense of height can be achieved which works for everyone. This is a problem with binaural techniques: when the HRTF's (head related transfer functions) do not match the individual listener, localisation will not work correctly. Moreover, if the spectral filtering would be used, then it would only be valid for a specific listening distance, as another distance, would mean that the source comes from a different elevation angle, and the filtering should change accordingly.

speaker arrays. Further research will of course be needed to evaluate whether this method leads to perceptually useful results.

2.6 Errors in the reproduction

2.6.1 Errors introduced for elevated source points

For point sources outside of the horizontal plane $y_0 \neq 0$. This means that the main contribution of this point should not be emitted from the speaker position, but from a higher or lower position. An error will be made as the actual propagation of the source signal will be from the speaker to the receiver point and not from the stationary point to the receiver point (see figure 2.5). This means that the amplitude will be higher and the delay will be shorter. Note that the error will be larger for sources further away from the horizontal plane and for sources closer to the array (for constant y_Ψ). This error may be compensated by adding a correction factor of

$$\frac{\tilde{\Delta r_0}}{\Delta r_0} e^{-jk(\Delta r_0 - \tilde{\Delta r_0})} \tag{2.32}$$

to the driver function of equation (2.30).

In figure 2.6 the errors in the delay and amplitude are shown for the source-speaker-listener path. The reference is the direct path from source to receiver. The errors are shown for the contribution of 1 speaker of the array, that is directly in between the source and receiver (i.e. the x-coordinates of source, speaker and receiver point are the same); the receiver point is at the reference line. The figure shows that the correction factor compensates the error.

2.6.2 Errors for receiver points not on the reference line

The derivation of the driver function (2.30) is valid for a reference line. For receiver points not on this line errors will be made that may further influence the interference pattern of the waves coming from elevated points.

For points closer to the array than the reference line, the vertical coordinate of the stationary point y_0 would be smaller than the one used for the reference line (see figure 2.7). Hence, r_0 should be larger for these receiver points and thus an error is made in delay and amplitude of the driver function. Δr_0 will be smaller for these points, resulting in further errors in the propagation delay and in the amplitude factor. For points that are further from the array than the reference line, the effects are opposite.

In figure 2.8a the errors are visualised for the contribution of one speaker. The figure shows the difference in delay and amplitude, when sound is propagated from source to receiver position. The reference line is taken at 3 meters from the array. In the delay we see that the error is in between -0.7ms and 0.25ms without the correction factor, and between -0.3ms and 0.15ms with the correction factor. These errors are up to 2.5 times higher than the delay error discussed in the previous section. Moreover, we see that the correction factor

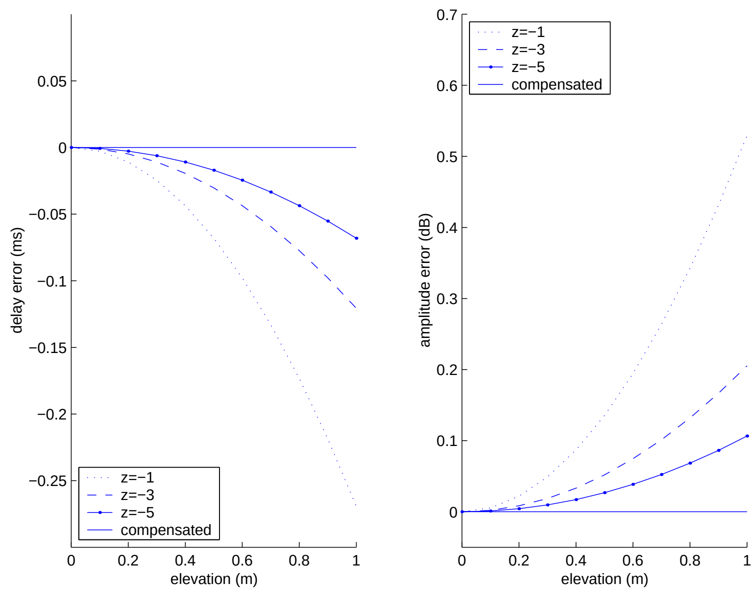


Figure 2.6: The delay and amplitude error for source points at different elevations and three different distances, z, behind the array. Note that the correction factor eliminates the error.

gives better results also for receiver points not on the reference line.

The amplitude error is up to 6 dB for the receiver distances plotted. This is an order of magnitude higher than the error described in the previous subsection. However, this error is mostly due to the attenuation errors that are made anyway as will be discussed in the next subsection. This is illustrated in figure 2.8b, were the attenuation errors for a source in the horizontal plane are plotted. For the elevated point, we see that the amplitude error is slightly larger for the calculation with the correction factor.

From these results it can be concluded that the correction factor should only be made for the delay, so the correction factor becomes solely:

$$e^{-jk(\Delta r_0 - \tilde{\Delta r_0})} \tag{2.33}$$

The effect of an error in the delay will be that the phases of the waves arriving at the listening point will not be correct and the interference between wave fields of different source points will be distorted; in the worst case certain frequency components may be amplified instead of attenuated through interference, causing a false frequency spectrum for the listener. The amplitude effects also influence the interference pattern.

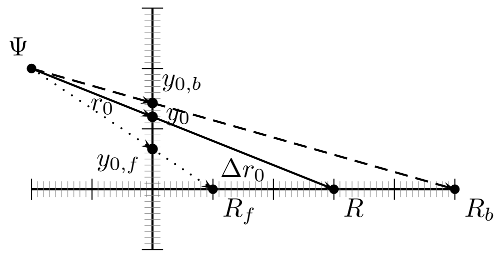


Figure 2.7: Geometrical explanation of the error made for receiver points not on the reference line. For receiver points R_f in front of the reference line R the vertical coordinate of the stationary point $y_{0,f}$ would be lower than y_0. For receiver points R_b behind the reference line R the vertical coordinate of the stationary point $y_{0,b}$ would be higher than y_0.

2.6.3 Attenuation errors

Sonke [Son00] elaborates on the attenuation error that is made, when using the $2\frac{1}{2}D$-operator, for source points in the horizontal plane.

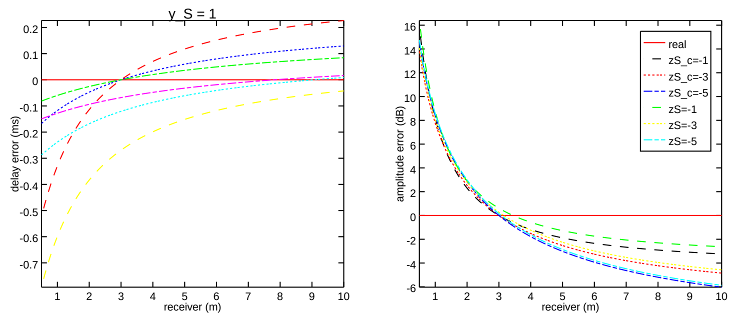
It can be seen that the amplitude rolls off with a factor that is just in between $\frac{1}{r}$, as would be necessary in a 3D situation, and $\frac{1}{\sqrt{r}}$, as would be in a 2D situation. The attenuation factor deviates from the 3D case at positions other than on the reference line. Between array and reference line, the amplitude factor is higher than the desired factor, on the other side of the reference line, the amplitude factor is lower. The order of magnitude of the deviation, as shown in figure 4.7 of his thesis, is however smaller than the amplitude deviation we have seen in the previous section; at least for receiver points further than the reference line.

Sonke also proposed some strategies to reduce the amplitude deviations. Both optimisations proposed are dependent on the desired receiver area and primary source location, which means that there is less flexibility of the system, which may be detrimental for realtime use.

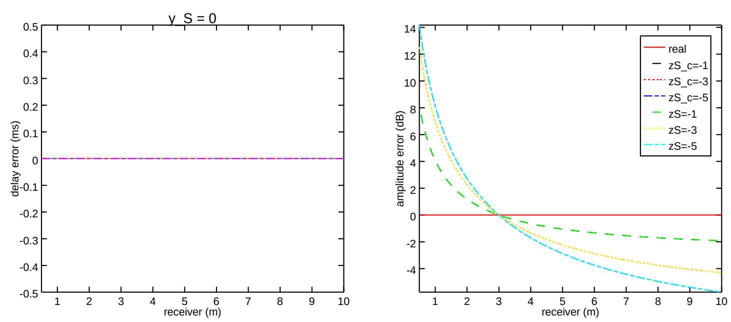
When working with a number of point sources, that are mutually dependent on (or correlated to) each other, as is the case with an arbitrary shaped source, and whose resultant wave field is a superposition of the wave fields of these point sources, attenuation differences in the reproduction play a more important role, than is the case for independent sources. As different parts of the source will in general emit different frequency spectra, the reproduced wave field will show differences in frequency spectra, possibly resulting in audible colouration of the sound, dependent on location in the listening area. This can be an undesired effect.

2.6.4 Spatial aliasing and truncation effects

In practice, the secondary sources on the boundary S_0 will not be continuous as has been assumed thus far, but instead will be realised by placing loudspeakers at discrete positions, i.e. the boundary will be spatially sampled. If the secondary source distribution is on the x-axis, we can write for the wave field $P(\vec{x}, \omega)$



(a) source at 0.5m above the horizontal plane



(b) source in the horizontal plane

Figure 2.8: The error in delay and amplitude for receiver points not on the reference line, which is at $3m$ from the array. Results with zS_C are with the correction factor, results with zS are without the correction factor.

within the listening area V [Spo05],

$$P(\vec{x},\omega) = \int_{-\infty}^{\infty} D(\vec{x}_0,\omega)V(\vec{x}-\vec{x}_0,\omega)dx_0 \tag{2.34}$$

where $V(\vec{x}-\vec{x}_0,\omega)$ denotes the wave field of the secondary sources, and $D(\vec{x},\omega)$ the driving function for the secondary sources. Now the $D(\vec{x},\omega)$ will be sampled, as it will be created by discrete loudspeakers; this can be written as a multiplication of D with a series of Dirac functions:

$$\begin{aligned} D_S(x,\omega) &= D(x,\omega)\frac{1}{\Delta x}\sum_{\mu=-\infty}^{\infty}\delta(x-\Delta x\mu) \\ &= \frac{1}{\Delta x}\sum_{\mu=-\infty}^{\infty} D(\Delta x\mu,\omega)\delta(x-\Delta x\mu) \end{aligned} \tag{2.35}$$

If we apply a spatial Fourier transform to equation (2.34), with (2.35) as the driving function, we get

$$\tilde{P}_S(\vec{k},\omega) = 2\pi \sum_{\eta=-\infty}^{\infty} \tilde{D}(k_x - \frac{2\pi}{\Delta x}\eta,\omega)\tilde{V}(\vec{k},\omega) \tag{2.36}$$

where the $\tilde{P}$ is the pressure in the wave number-frequency domain, k_x is the spatial frequency in the x-direction, and η is an index.

The spectrum of the driving function $\tilde{D}_C(k_x,\omega)$ is dependent on the virtual source to be reproduced. For a plane wave aliasing in the reproduced wave field will be present if the frequency spectrum of the signal to be reproduced contains frequencies that do not fulfill the aliasing condition

$$f \leq \frac{c}{\Delta x(1+|\sin\alpha_{\mathrm{pw}}|)} \tag{2.37}$$

where α_{pw} is the incidence angle with the normal on the speaker array of the plane wave. For frequencies above this aliasing frequency, there will be components present in the wave field with incidence angles:

$$\sin\alpha_{\mathrm{pw},\eta_{\mathrm{al}}} = \frac{\frac{2\pi}{\Delta x}\eta_{\mathrm{al}} + \frac{\omega}{c}\sin\alpha_{\mathrm{pw}}}{\frac{\omega}{c}} \tag{2.38}$$

η_{al} is the counting index from equation (2.36), for which the following condition is true:

$$|\frac{2\pi}{\Delta x}\eta_{\mathrm{al}} + \frac{\omega}{c}\cos\alpha_{\mathrm{pw}}| \leq \frac{\omega}{c} \tag{2.39}$$

and $\eta_{\mathrm{al}} \in \mathbb{Z} \setminus 0$.

Spatial aliasing for point sources

The anti-aliasing condition (2.37) was derived for a plane wave. For the reproduction of point sources, the situation will be slightly different: the incidence angle of the wave front will be different for each loudspeaker and thus the highest frequency for which there is no aliasing is present will be slightly different (this case is the reverse of incidence of a plane wave on a circular array, which is discussed below). Consequently, if the source signal does contain frequencies which harm the anti-aliasing condition, the artefacts will be present in the reproduced wave field, their local strength being dependent on the location of the virtual point source. This means that for certain locations in the listening area the reproduction may be (almost) correct, while for other locations in the listening area, there are artefacts audible.

To illustrate this effect, a simulation was made utilising *Matlab* [Mat] for a loudspeaker array of 10 meters long with a speaker distance of 20cm., and the resulting signals at a distance of 3 meters from the array were calculated. The resulting frequency response is shown in figure 2.9. It can be seen that the aliasing will be heard relatively stronger when moving further away from the source position.

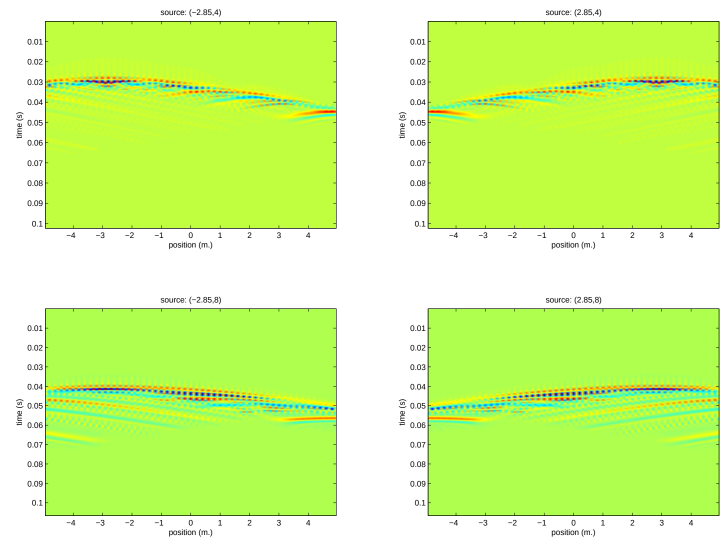
Circular and arbitrary shaped arrays

From Spors' [Spo05] discussion on spatial aliasing with circular arrays, we know that for the reproduction of plane waves, there will always be aliasing artefacts present. Two conclusions are drawn: first: the higher the bandwidth of the plane wave, the more energy will be contained in the aliasing contributions of the reproduced wave field. Secondly, the further the listener position is from the active secondary sources, the lower the energy of the aliasing contributions.

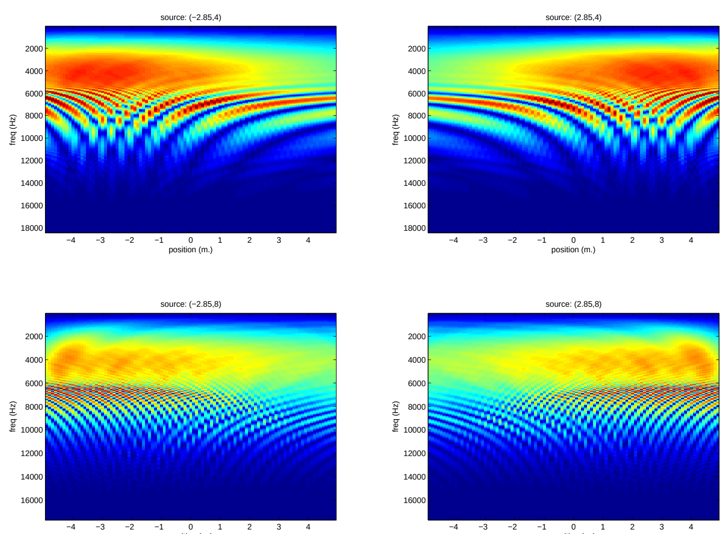
The case of a circular array, can be generalised to an array with arbitrary shape, by mapping the boundary ∂V of the arbitrary listening area V to a circular boundary $\partial V'$ of a listening area V'. According to the Riemann mapping theorem [Wei03] every simply connected region can be mapped with a one-to-one transformation to a unit circular region using an analytic function. The mapping of an equidistantly sampled arbitrary shaped array onto a circular array may result in a non-equidistant angular sampling on the circle. As a consequence, it will not be possible to derive a generic anti-aliasing theorem.

Truncation effects

In practice the loudspeaker array will also be finite and truncation effects will occur [Sta97, Son00]. A spatial taper over the driving function can reduce the perceptual disturbing cues created by the truncation. By applying such a taper, the diffracted waves are smeared out in time and place. The taper also influences the correctly synthesised wave field, so the choice of a taper is always a compromise between the reduction of the diffraction artefacts and the size of the listening area.



(a) time domain



(b) frequency domain

Figure 2.9: Illustration of spatial aliasing for a point source. On the left-hand sources at location with $x = -2.85$, on the right-hand with $x = 2.85$. The upper plots are at a distance of $1m$ behind the array, the lower plots at $5m$.

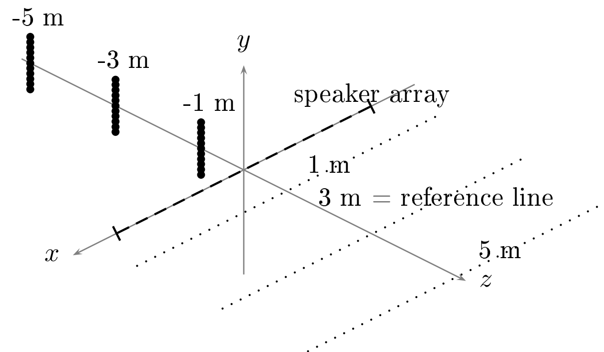


Figure 2.10: Geometry of the simulations. Source points are sampled at 0.1m intervals in y-direction. The speaker array and the receiver lines are sampled at 5cm intervals. The speaker array goes from -3 to 3m, the receiver lines from -3.5 to 3.5m in x-direction.

2.7 Simulations

2.7.1 Setup

A combination of point sources at different elevations, running from -0.5 to +0.5 m in steps of 0.1m, centred behind the WFS array, was used as the source for simulations done with *Matlab*. The source signal was a Gaussian wavelet with no frequency components above 750Hz. The source signal was extrapolated with a wave field extrapolation operator based on the Rayleigh I integral. In both simulations, the following source and receiver distances were used: 1, 3 and 5 meter behind the array and 1, 3 and 5 meters in front of the array respectively (see figure 2.10). The reference line was at 3 meters in front of the array, parallel to it. In order to avoid truncation effects, a Hanning window over 20% of the array length was used on both sides (see [Ver98]).

The correct field[2] was compared with WFS reproduction followed by extrapolation from the speaker array to the listening position. The WFS reproduction array was sampled at 5cm intervals[3], as were the listening positions. The corresponding aliasing frequency for this array is 3.4kHz, thus we can assume that the WFS reproduction does not introduce aliasing problems.

2.7.2 Results

In figures 2.11 to 2.13, the time domain responses are shown for a source at an elevation of $0.5m$, at different distances behind the array and for different receiver distances. The reproduction seems to be quite good, though we can see that for the WFS reproduction there are artefacts as the source is further away, but these are the same for a source without elevation (not shown here), so these

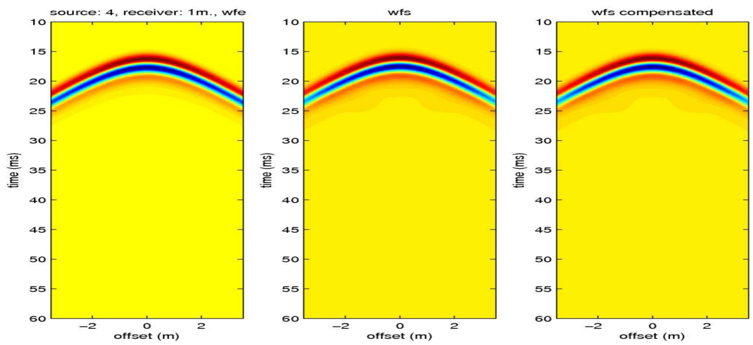
[2] calculated with direct wave field extrapolation from source to listening position

[3] this may not be realistic, but as the aim is to look at the effect of elevated points, the distance was chosen in order not to introduce aliasing artefacts

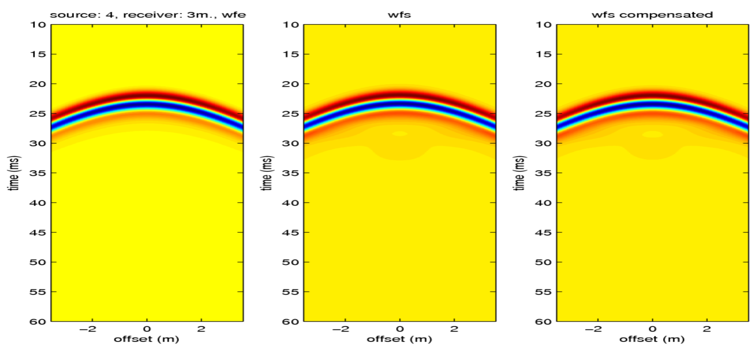
can be attributed to truncation artefacts.

The difference between the WFS reproduction with and without the correction factor is negligible.

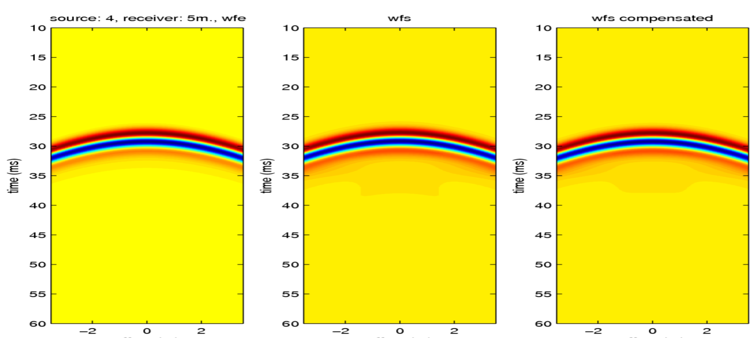
The corresponding frequency responses are shown in figure 2.14 to 2.16. Here we can see that the artefacts seen in the time domain have an influence on the frequency response: at certain positions in the listening space some frequencies will be stronger than at other positions.



(a) At a receiver distance of $1m$ from the array

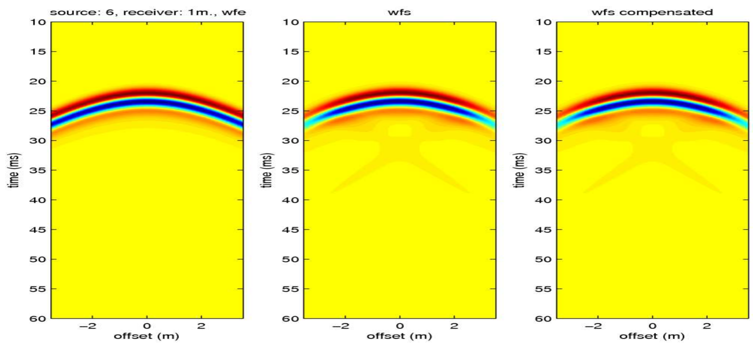


(b) At a receiver distance of $3m$ from the array

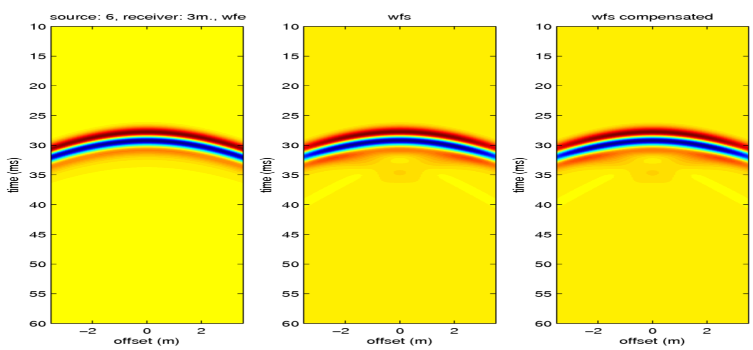


(c) At a receiver distance of $5m$ from the array

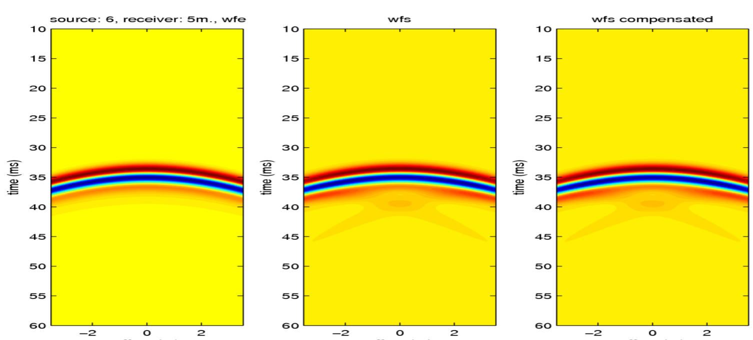
Figure 2.11: Source at $1m$ behind the array, with an elevation of $0.5m$, in the time domain. Left-hand: correct field (wfe), middle: WFS reproduction with driver function (2.30) and right-hand: WFS reproduction with the compensated driver function ((2.30) with (2.32)).



(a) At a receiver distance of $1m$ from the array

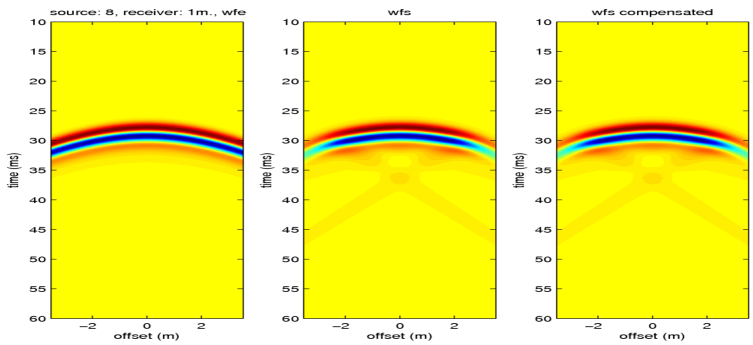


(b) At a receiver distance of $3m$ from the array

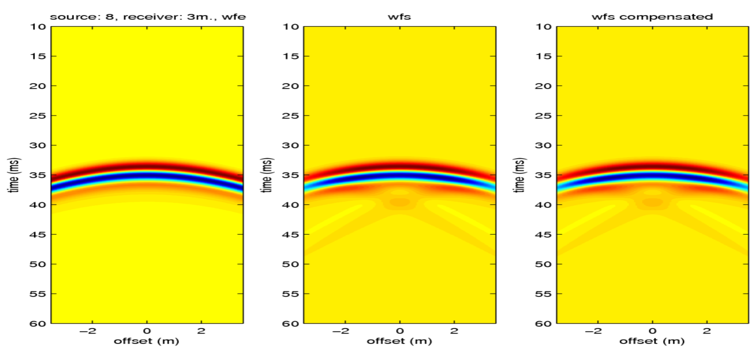


(c) At a receiver distance of $5m$ from the array

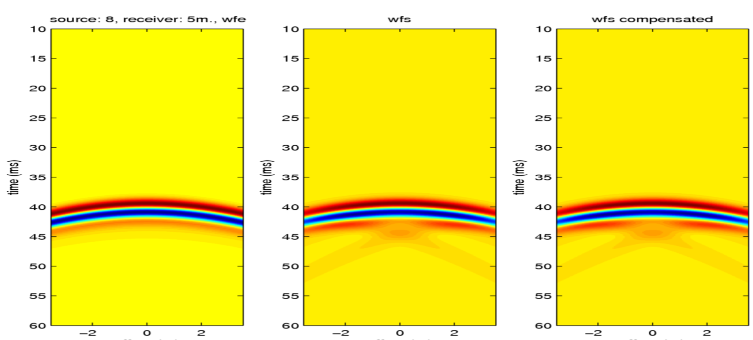
Figure 2.12: Source at $3m$ behind the array, with an elevation of $0.5m$, in the time domain. Left-hand: correct field (wfe), middle: WFS reproduction with driver function (2.30) and right-hand: WFS reproduction with the compensated driver function ((2.30) with (2.32)).



(a) At a receiver distance of $1m$ from the array

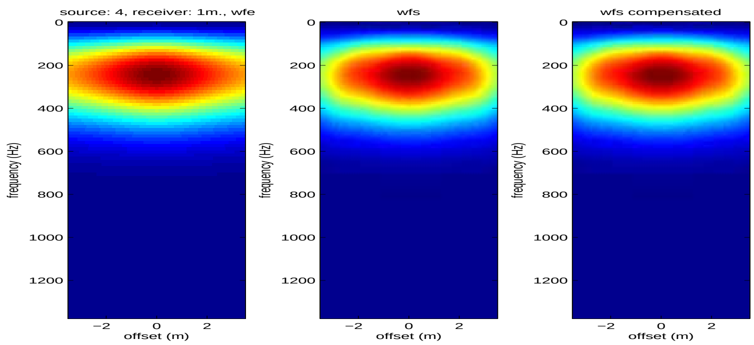


(b) At a receiver distance of $3m$ from the array

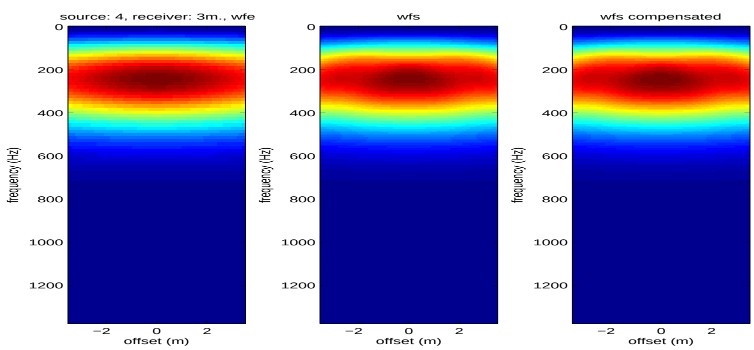


(c) At a receiver distance of $5m$ from the array

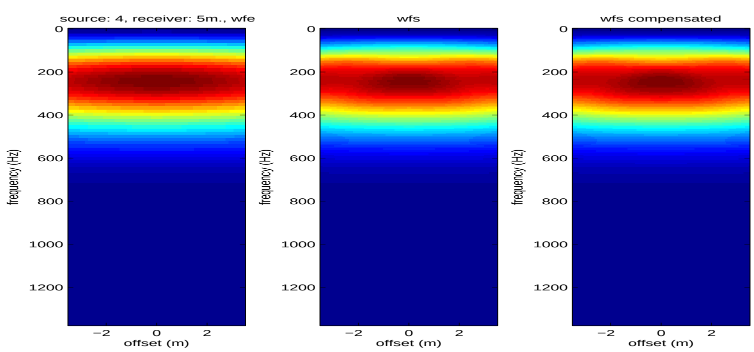
Figure 2.13: Source at $5m$ behind the array, with an elevation of $0.5m$, in the time domain. Left-hand: correct field (wfe), middle: WFS reproduction with driver function (2.30) and right-hand: WFS reproduction with the compensated driver function ((2.30) with (2.32)).



(a) At a receiver distance of $1m$ from the array

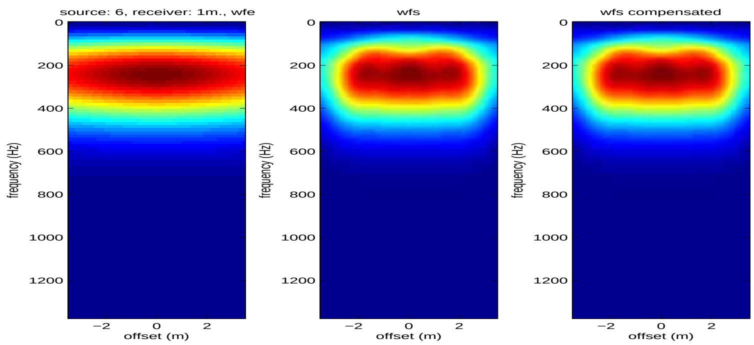


(b) At a receiver distance of $3m$ from the array

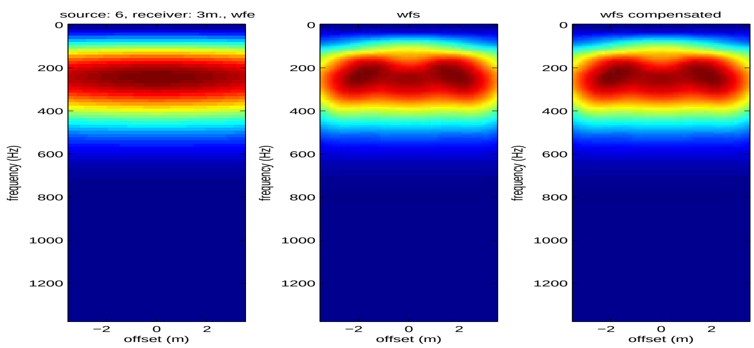


(c) At a receiver distance of $5m$ from the array

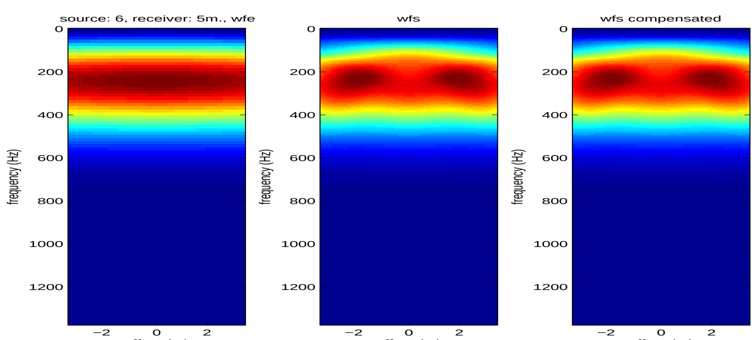
Figure 2.14: Source at $1m$ behind the array, with an elevation of $0.5m$, in the frequency domain. Left-hand: correct field (wfe), middle: WFS reproduction with driver function (2.30) and right-hand: WFS reproduction with the compensated driver function ((2.30) with (2.32)).



(a) At a receiver distance of $1m$ from the array

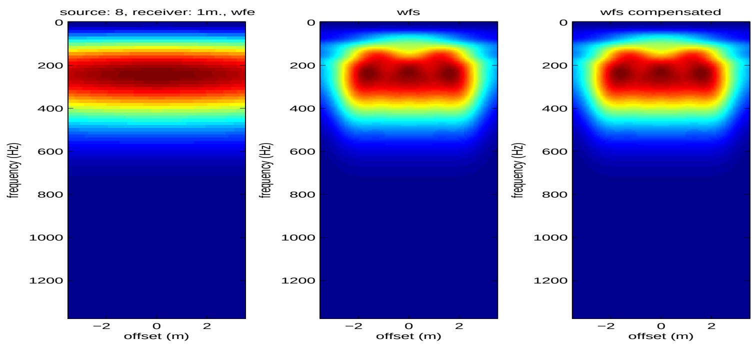


(b) At a receiver distance of $3m$ from the array

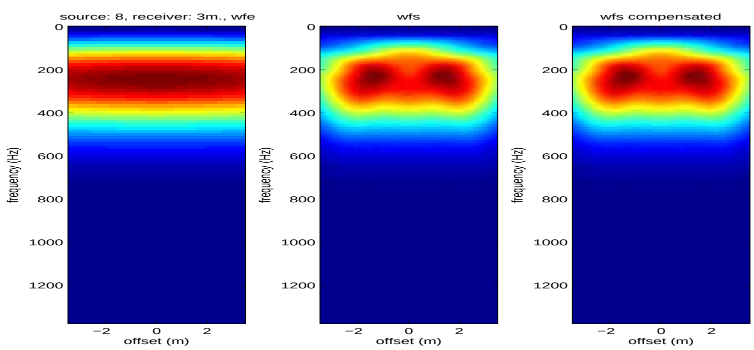


(c) At a receiver distance of $5m$ from the array

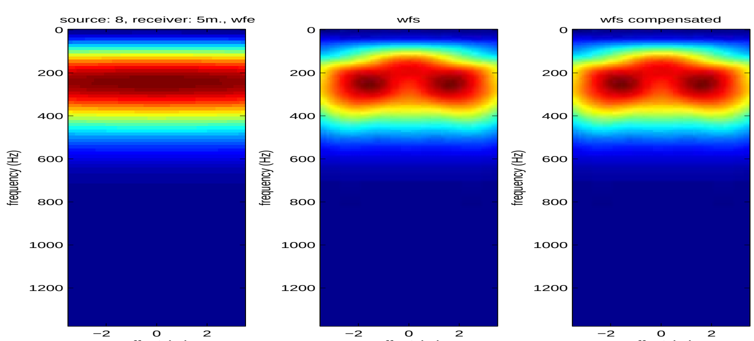
Figure 2.15: Source at $3m$ behind the array, with an elevation of $0.5m$, in the frequency domain. Left-hand: correct field (wfe), middle: WFS reproduction with driver function (2.30) and right-hand: WFS reproduction with the compensated driver function ((2.30) with (2.32)).



(a) At a receiver distance of $1m$ from the array



(b) At a receiver distance of $3m$ from the array



(c) At a receiver distance of $5m$ from the array

Figure 2.16: Source at $5m$ behind the array, with an elevation of $0.5m$, in the frequency domain. Left-hand: correct field (wfe), middle: WFS reproduction with driver function (2.30) and right-hand: WFS reproduction with the compensated driver function ((2.30) with (2.32)).

Chapter 3

System setup

3.1 Hardware setup

At the TU Berlin Weske [Wes01] set up a 24-speaker system using active loudspeakers (FOSTEX 6301B) (see figure 3.1), a Linux PC with an RME Hammerfall Digi9652 soundcard, and Marian Adcon converters. The speaker distance is 12.5 cm.

On this system, the software *sWONDER* was initially developed and used for the composition of several pieces, described in section 4.3.

In 2006/2007, the TU Berlin launched a project to equip one of the lecture halls with a large WFS system [MGM+07, BAM07], of in total 832 loudspeaker channels, both for sound reinforcement during the regular lectures, as well as to have a large scale WFS system for both scientific and artistic research purposes. The loudspeakers are built into loudspeaker panels [GMMW07], each providing 8 audio channels, which are fed with an ADAT signal. The 8 channels are each played back with 3 loudspeakers, that each have a different filtering of the signal. Looking at figure 3.2, the first row of speakers underneath the 2 larger speakers, that are used for the frequencies up to 200 Hz, emits up to 5.5 kHz, the 2nd row up to 10 kHz, and the third up to 17 kHz. The two larger speakers emit the low-pass filtered sum of the 4 channels below it. Within the speaker panel, the signal is filtered with a digital filter for (linear phase) equalisation, which can be reprogrammed if necessary. The filtering for the different rows of speakers is achieved by analogue filtering (i.e. capacitors). The panel was designed to have a broad horizontal radiation angle, and a narrow vertical radiation angle, so that floor and ceiling reflections are avoided.

To drive these speakers a cluster of 15 Linux computers is used. Each computer calculates the loudspeaker signals for 56 loudspeaker channels. Each computer is equipped with an RME HDSP MADI [RME] sound card. Each MADI output is connected to an MADI to ADAT bridge (RME ADI648), which is mounted inside the wall, so that the ADAT cables can be kept relatively short (up to 10 meters). The input to the system is multiplexed to each MADI sound card with the use of MADI bridges (RME MADI Bridge).

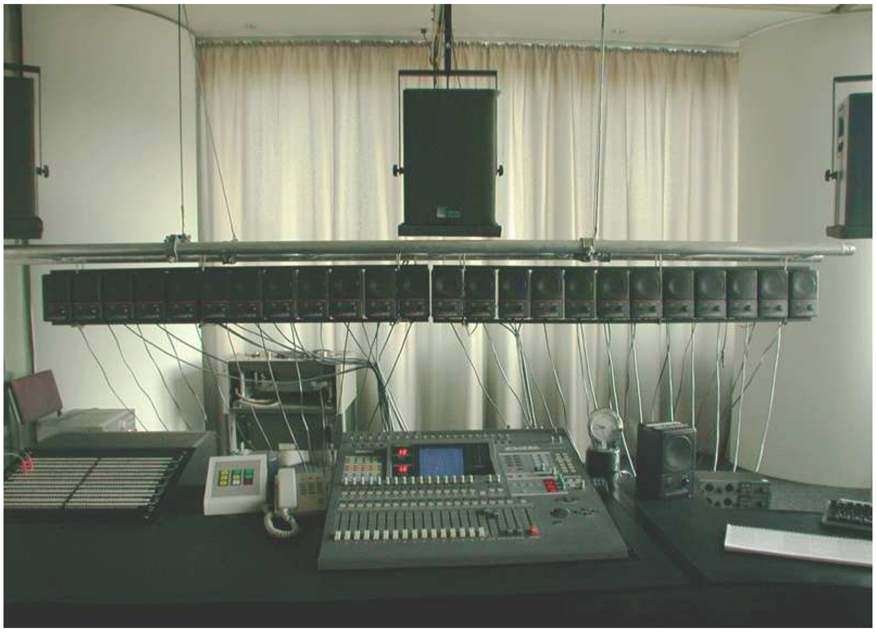


Figure 3.1: The 24 FOSTEX loudspeakers in a WFS setup in the electronic studio of the TU Berlin *(photo: Folkmar Hein)*.

Figure 3.2: The loudspeaker panel for the lecture hall at the TU Berlin.

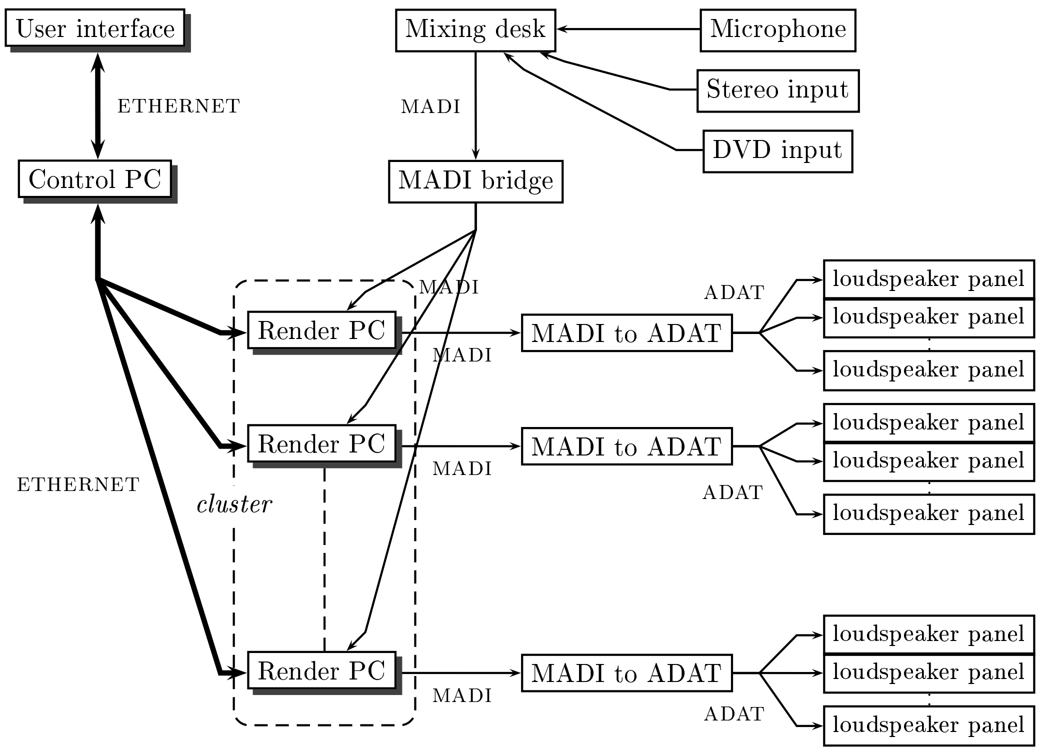


Figure 3.3: Schematic overview of the hardware setup for the WFS system in the lecture hall of the TU Berlin.

The cluster has two networks, one for the OSC [WFM03] communication, and one for data-transfer. Separating these network functions, ensures that the OSC communication is fast. The master machine (Control PC) acts as a bridge to the outside world and is the only computer that is connected to an external network.

A general overview of the hardware setup is given in figure 3.3 and a photograph of the lecture hall is shown in figure 3.4. Note that the lecture hall has a slight elevation when going from the stage to the back of the hall, which means that the speaker array needed to make a similar elevation when going to the back of the room. This resulted in several adaptations in the software to ensure that the sound reproduction is correct, which will be discussed below. Other architectural constraints for mounting the speaker array were: the necessity to have a blackboard in the front (causing a high elevation above the stage in the front), control room windows in the back (thus the speaker array had to be mounted quite low, at the height of the seats of the back row), doorways on the side (another reason for the elevation in the front, and some missing speaker

Figure 3.4: Panorama view of lecture hall H0104 of the TU Berlin, equipped with an 832-channel WFS system *(photo: Folkmar Hein)*.

panels in the back), as well as esthetic demands. The design of the speaker panels had to be both robust (against vandalism) and maintainable.

The *sWONDER* software was adapted to control this system, in such a way, that it can also be used by similar but not necessarily identical systems.

3.2 Software

The software *sWONDER*[1] [BP04, Baa04, Baa05], was developed with the aim to provide composers with an easy to use interface to use Wave Field Synthesis for the composition of spatial movements.

Initially a composition tool only [Baa03], the program has developed to become a fully capable WFS software that can be used both for small and large WFS systems, which need several computers to render audio [BHSK07].

The software is divided into several parts[2]:

- a graphical user interface,
- a score player/recorder,
- a control unit,
- a realtime time domain render unit,
- a realtime frequency domain render unit,
- an offline render unit,
- a timer application *jfwonder*,

[1]an acronym for *sound Wave field synthesis Of New Dimensions of Electronic music in Realtime*. The software is released under the GPL license at `http://swonder.sourceforge.net`. The original name for the software was WONDER, but that name was already taken on SourceForge by a project for WebObjects applications, so a slight name change had to be made. *To swonder* is yet to be recognised as an English word for *to move freely in space*.

[2]The software described in this chapter has been developed by a team of programmers, coordinated by the author. Simon Schampijer programmed the control unit, Torben Hohn the audio rendering units, as well as the LADSPA plugins, Daniel Plewe the score player/recorder, Eddie Mond the graphical user interface and Sebastian Roos programmed the VST plugin. Furthermore, Thilo Koch installed the cluster and networking.

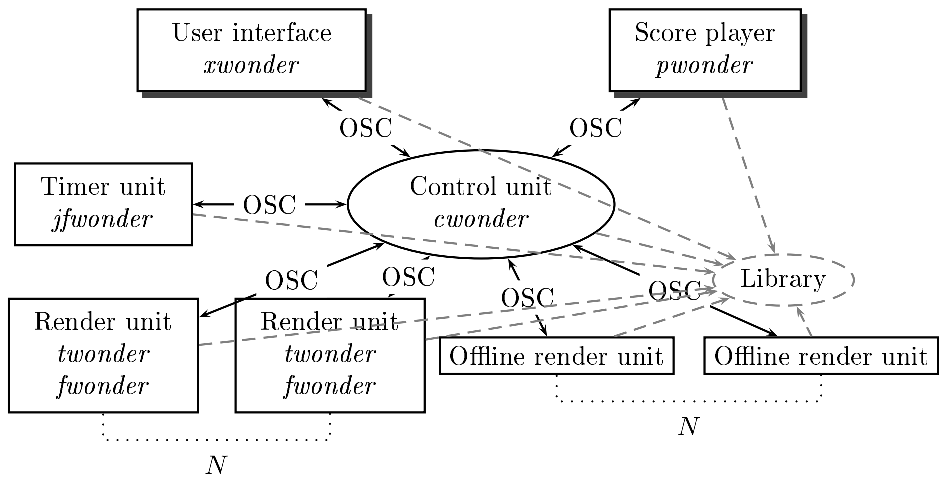


Figure 3.5: Schematic overview of the different parts of the software *sWONDER*. The control unit can communicate with an arbitrary number (N) of realtime and offline renderers.

- and a common library for general functions.

Communication between the different parts of the program is based on the OSC protocol [WFM03]. Figure 3.5 gives an overview of the program parts and their communication.

Graphical User Interface

A graphical user interface has been developed primarily for enabling an easy interface to create and choose scenes, as well as to create, edit and visualise scores of movements [Mon07]. The details about this are described below.

Score player/recorder

The system can take any kind of audio input, so that the user can use the audio player (s)he prefers to play the audio. The score player/recorder is used to synchronise with an audio player and record and playback source movements. Below is a detailed description.

Offline renderer

For room simulations or for arbitrarily shaped sound sources (see chapter 6), the calculations for the impulse responses for each speaker can take quite long, and cannot be performed in realtime, so an offline render unit is needed, which can utilise the cluster, benefiting from parallel execution.

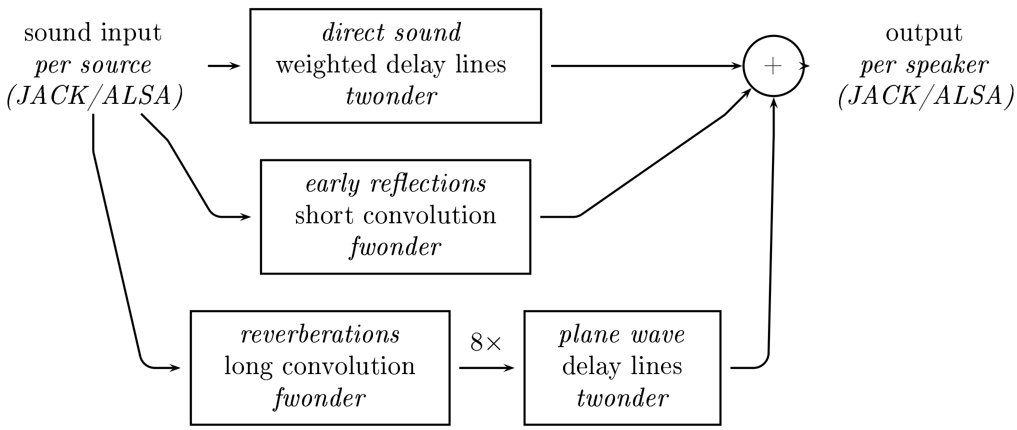


Figure 3.6: An overview of the audio signal processing by the realtime render unit.

Control unit

The control unit (*cwonder*) acts as a bridge between the user interface and the audio renderers; it also communicates with the score player/recorder. Though the *sWONDER* suite of programs supplies a graphical user interface, any other program that can send (and receive) OSC can be used to control the system. The user interface only needs to communicate with the control unit, and does not need to know anything about the audio rendering details; the control unit takes care of that. The control unit is described in more detail below.

Rendering engine

The realtime render unit is responsible for the actual audio signal processing. It has several ways to deal with the audio streams: playback of direct sound, utilising weighted delay lines, convolution of the input sound for early reflections, and convolution of the input sound for reverb followed by weighted delay lines to create plane waves with the reverb tail. Schematically this is shown in figure 3.6.

The rendering engine consists of two parts: *twonder* for the delay line implementation, and *fwonder* for the convolution. Both programs are controlled by OSC and are described below; audio input and output have the audio server JACK [Dea] as the audio back-end.

Plugins

To automate movements of sources with a multi-track sequencer, LADSPA plugins [Fur] were created to do so. One to send out OSC movement commands for one source, and one to send out OSC movement commands for a group of

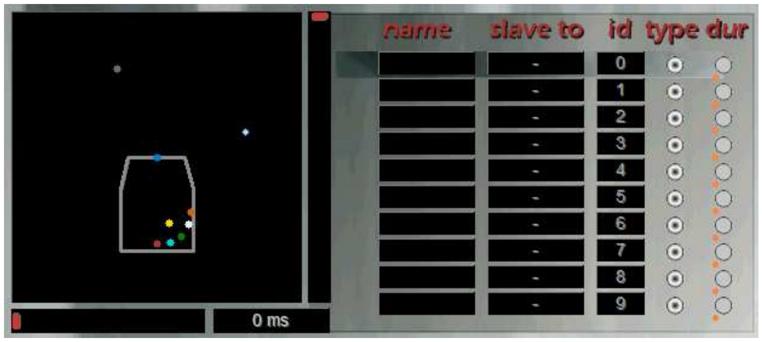


Figure 3.7: Screenshot of the VST plugin for *sWONDER*.

two sources and one for a group of four sources, of which a centre point can be moved, and the scale and rotation of the cloud of points can be varied (figure 3.8).

A VST plugin [VST] has also been developed (figure 3.7).

3.3 The control unit - *cwonder*

sWONDER is a "distributed application" that communicates over OSC. *cwonder* is the glue part that each of the other parts communicates with. The only thing the other parts need to know is the address of *cwonder* which can be retrieved from a configuration file. The other parts, or other applications, can connect to different streams that *cwonder* sends out:

- `/WONDER/stream/render` contains all information to render the audio for the sources.
- `/WONDER/stream/score` contains all information to record and play the source movements.
- `/WONDER/stream/visual` contains all information to display the status of all the sources and settings.
- `/WONDER/stream/timer` gives a regular update on the current time of the system.

When connected to a stream, *cwonder* sends out regular messages ("ping") to which needs to be replied (with "pong"). If a program that is connected does not reply for a certain amount of time, *cwonder* will stop sending data to that program.

An application can send to a stream directly as well. This message is then forwarded to a stream without *cwonder* processing it. One case where this should be used is for replies to actions. If for example playback of a score is started from the user interface the score player should use the following syntax to reply to this command:

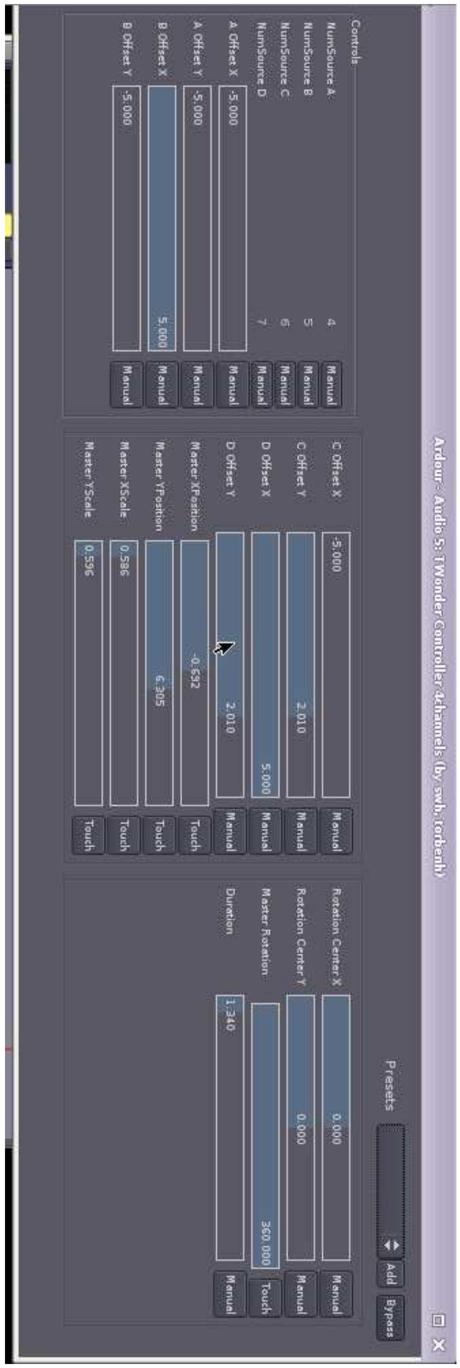


Figure 3.8: The 4 channel LADSPA plugin in Ardour.

`/WONDER/stream/score/send` ssis `/WONDER/reply` path 0 "start to play"

The namespace of the message is `/WONDER/stream/score/send`. The message the other receivers of this stream will receive is `/WONDER/reply` in this case which is the first argument of this message. The other arguments just follow like in the other cases. So *cwonder* will create from the above message the following message and send it to the receivers of the stream score:
`/WONDER/reply` sis path 0 "start to play"

3.4 Rendering - *Direct sound*

The direct sound of a WFS synthesised source, consists of the delayed and attenuated source signal. This delay and attenuation is unique for each speaker. The direct sound of the source is rendered in the time-domain by the component *twonder*.

3.4.1 Point sources

In order to calculate the delay and attenuation for a point source the following steps are taken in the calculation:

1. calculate the vector between source and speaker,
2. calculate the in-product of this vector with the normal vector on the speaker,
3. if this normal vector is negative, and if the source is inside the array of speakers (check whether it is within a polygon defined by the speaker locations)
    (a) check whether the source is within a maximum distance from the speaker, if it is then apply a window within a "window width" for a smooth fade out,
    (b) do a recalculation of the distance (delay time) based on a 3D vector calculation (important for rooms with an elevation),
    (c) set the distance to a negative value,
4. adjust delay to smallest delay (to prevent illegal assignment later on in the program),
5. check whether distance is bigger than 1.5 times the speaker distance:
    (a) *bigger and behind speakers* calculate the corresponding factor,
    (b) *bigger and in front of speakers* calculate the corresponding factor,
    (c) *smaller* calculate factor as an interpolation between the factor at 1.5 times speaker distance behind and in front of the speakers,
6. multiply the factor with the window and $\cos\phi$ with the speaker.

The factor of 1.5 the speaker distance ensures that even if the source moves through the speakers right in between two speakers, there will always be some speakers that play sound, and the amplitude will not "explode" (note that in Δr in equation (2.30) becomes 0 at the speaker position).

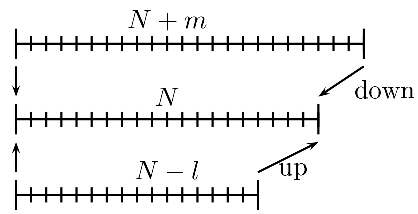


Figure 3.9: Illustration of the resampling problem: if the delay time gets longer within a certain block, we need to output more samples than we have available in our buffer. Thus, we need to upsample the available samples. If the delay time gets shorter, we need to output less samples than we have available in our buffer and we need to downsample them.

3.4.2 Delay lines

To initialise the delay lines, the length of the delay lines need to be determined. The length is related to the largest distance a source will have to a speaker. Also, it needs to be decided how far in front of the speakers we want to move a source, as this determines the needed delay offset. If no focused sources are needed, we can set the delay offset to a smaller number, thus introducing less latency in the system. These options can be set per source.

3.4.3 Moving sources

When a source moves, the delay time will change continuously, as well as the volume factor. In *twonder* the delay times and factors for the start and the end of the block are calculated (thus these are a kind of anchor points), and the samples inside the block are resampled. This is clarified in figure 3.9. If the delay time is 20 samples at the start of the block, and 30 samples at the end of the block, we need to output 10 samples less than the actual block size N. Thus, we need to resample N to $N-10$ samples. Because of the CPU restraints (we need to do this for a lot of delay lines in realtime), an efficient resampling algorithm is needed. A linear interpolated resampling was chosen. The implementation is a modified version of Bresenham's line drawing algorithm [Wik07], which eliminates the need to cast a float to an integer in the inner loop; this is an advantage as in most CPUs there is a separate integer and float computation unit; casting a float to an integer causes a hold-up in both calculation pipelines, so should be avoided in intensive time-critical calculations.

Moving a source in this way, creates a Doppler effect, which will be audible if the movement is very fast. Sometimes it is not desired (by the composer) to hear a Doppler effect, so another option for movement is provided, which we have called a *fade jump*. Using this option, the illusion of movement is created by fading the source out on one position, while fading it in on the next position. The user can set a threshold at which the system will stop resampling and switch to blockwise crossfades. The threshold is the number of samples which would need to be added or removed due to the movement of the source. This is per calculation block, and thus depends on the calculation block size the system is configured to.

3.4.4 Plane waves

Plane waves are achieved by varying the delay times for each speaker, based on the angle the wave front makes with the speaker array. A delay offset is created by giving the plane wave a point of origin in space, in addition to its direction. This approach also makes it possible (in principle) to switch from a point source to a plane wave and vice versa.

The amplitude is adjusted by a factor depending on the angle the plane wave makes with the speaker array, and with a factor to compensate for the difference in loudness with a point source. This compensation factor can be adjusted in the configuration at startup by the user.

Plane waves can be used to simulate sources that are very far away and only have a direction, or to simulate reflections, as will be described in the next section.

3.4.5 Corrections for elevation angle of the speaker array

The lecture hall H0104 has a slight elevation. This is made clear in figure 3.10, where the coordinates of the outline of the hall are shown. From 8.8 meters measured from the blackboard at the front wall, the speaker array makes an elevation, ending up 4.4m. higher in the back, over a distance of 16.9m., than it is in the front.

For a focused source, it is important that the waves converge from the speaker to the intended source position, and for this the actual path length of the waves to the source position needs to be taken. This means that the z-coordinate of the speakers cannot be neglected. In the source code this problem was solved by calculating an appropriate z-coordinate for the source, so that it is always placed in the plane of the speakers. The delay times for each speaker are then calculated based on the 3D-coordinates of the virtual source and the relevant speakers.

3.5 Rendering - *Room simulation*

Room simulation is achieved by adding reflections to the direct sound. This can be achieved in several ways:

1. inclusion of a first reflection in the delay line,
2. doing a short convolution for early reflections for each speaker with offline calculated impulse responses (IRs), and
3. doing a convolution with a longer impulse response, the result of which will be played back using plane waves from different directions.

In the first option a filter on the reflected sound can also be included, provided the filter can be created with ca. 8 FIR taps.

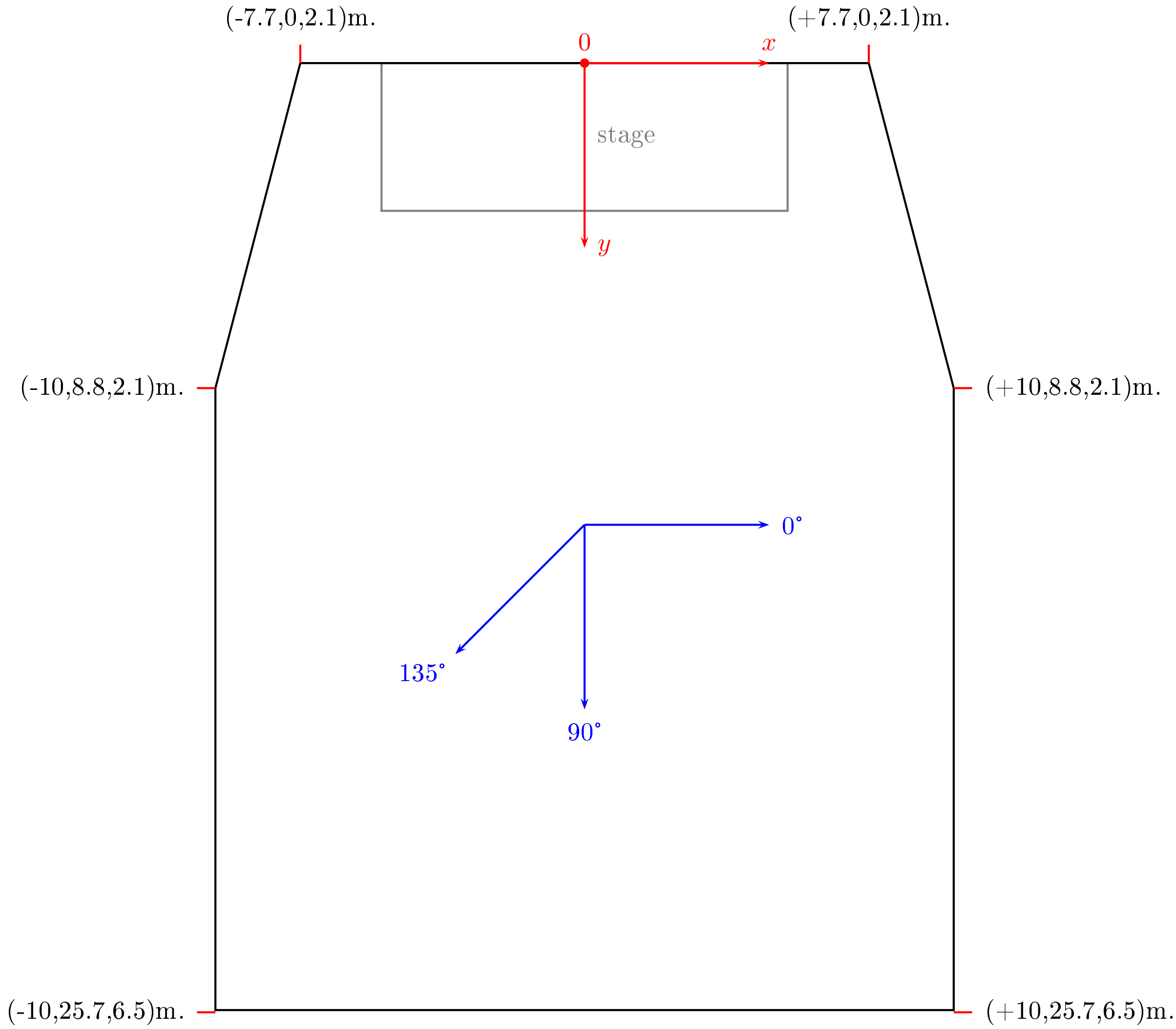


Figure 3.10: The coordinates for the lecture hall H0104.

A convolution engine, *fwonder*, has been developed to enable the second and third option, but the offline renderer to calculate the early reflection impulse responses is not ready yet. Instead, other methods could be used to calculate the early reflection IRs, such as an old version of *sWONDER*, or using an approach based on measurements such as described in [MdV06, dVLM06, HdV01].

In the case of early reflections, the impulse responses are unique to each source position and speaker. Thus for each speaker a convolution needs to be made. This option is CPU-intensive and memory-intensive, and a clever IR caching is needed to enable the use of a lot of impulse responses. In [Hel03] research is presented from which can be concluded how closely grid points need to be spaced to ensure perceptual consistency of the wave field, for a specific setup (depending on the dimensions of the virtual room, as well as the size of the desired listening area).

Option 3 can be used for reverberation. Research at the TU Delft has shown that using 8 plane waves (at 45 degrees interval directions) is sufficient to create a realistic reverberation [SdV97].

3.5.1 Convolution

The *fwonder* program implements a fast convolution from multiple inputs to (even more) multiple outputs. It uses the same complex multiplication method as BruteFIR [Tor05]. This method consists of reordering the filter coefficients in such a way, that the CPU needs to fetch new data less often from memory, which speeds up the computation process.

Instead of extending BruteFIR, a convolution engine was rewritten from scratch, because this was considered faster than extending BruteFIR, due to the lack of transparency and documentation of the BruteFIR code. Other available solutions were not written in C++ or tied to SuperCollider [Ker06], which would have slowed down development also. So it was decided to reimplement the algorithm, while learning from the other implementations.

The convolution engine *fwonder* is also capable of non-uniform partitioned convolution as described by Sommen [Som89].

3.5.2 IR caching

When a source is moving, the active impulse responses need to be changed. Because the set of impulse responses does not fit into memory, a cache structure needs to manage the loading of the impulse responses from disk.

This problem is solved as follows: when the position of a source changes the UI sends the absolute position in meters to the control unit. The control unit sends the new position to *twonder*, and simultaneously calculates the corresponding (closest) grid position for which an early reflection impulse response is available, and sends this information to *fwonder*. *fwonder* then switches the impulse responses used in the convolution to the new ones. Crossfading is used to reduce the artefacts of this process.

Because the loading of new impulse responses is a task that takes some time to complete, it should happen before an event actually occurs if possible. In realtime mode it is not known in advance what parameters of which source will change next. As a solution the grid of points for which IRs are calculated is divided in anchor points and normal points. Anchor points are points whose IRs are always stored in memory. When a source moves to a new location, first the IR of the anchor point is used, and then the surrounding points are loaded into memory, so that changes to locations nearby can be made in realtime (see figure 3.11a). When there is a score, we know which IRs are needed in the future, and we can determine the needed IRs in time, as shown in figure 3.11b.

The control unit takes care of this scheduling of loading and unloading of IRs and sends commands to the render units to perform this (i.e. the render unit *fwonder* is 'stupid' and just follows the orders of the control unit).

3.5.3 Calculating the IRs

The IRs as described above, will need to be calculated beforehand with the offline renderer. This is handled as follows: in the UI the user defines a grid of points and virtual room dimensions. Then he sends a message to the control unit to start the calculation. The control unit then communicates with all the offline renderers that are running, to perform this task, and sends a message back to the UI when the task is completed. Afterwards, the calculated IRs can be used in realtime.

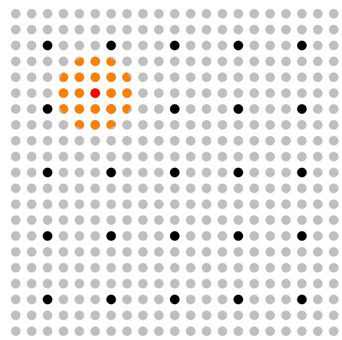
3.6 Score playback and recording

In order to playback the movements of various sources a score player and recorder is needed that can synchronise with an audio player. There are several clocks available for synchronisation, the most common being supported is MTC, which stands for Midi Time Code. On the Linux platform JACK Transport [Dea] is commonly used, which would be another option.

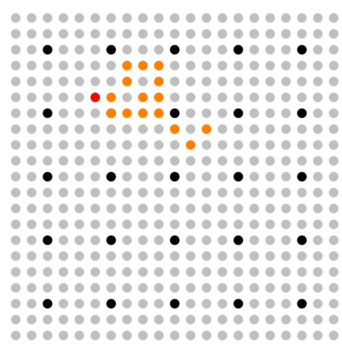
In order to record a score, recording needs to be turned on, so that *cwonder* sends the score the source data as they change over time. Further flags can be set to record only specific sources, while playing back the movements of previously recorded sources.

The source movements are stored, as the OSC messages come in: the score recorder notes the timecode from the clock, and stores the OSC message. If the OSC message contains the same information for a source as the previous message, the message is discarded. This results in an *event* based recording, which is of benefit as less information is stored. However, it makes jumps in time more complicated to handle.

If a user jumps to another point in time in the composition, then the score player needs to determine the locations of the sources at that moment in time. In principle, the score player would need to parse all data from the start in order to determine the information at the chosen point in time. In order to speed up this jump, the score player can create anchor points in time, for which it knows the current locations, so that when a time jump occurs, it only needs to parse the data from the anchor point which is just before the requested point in time.

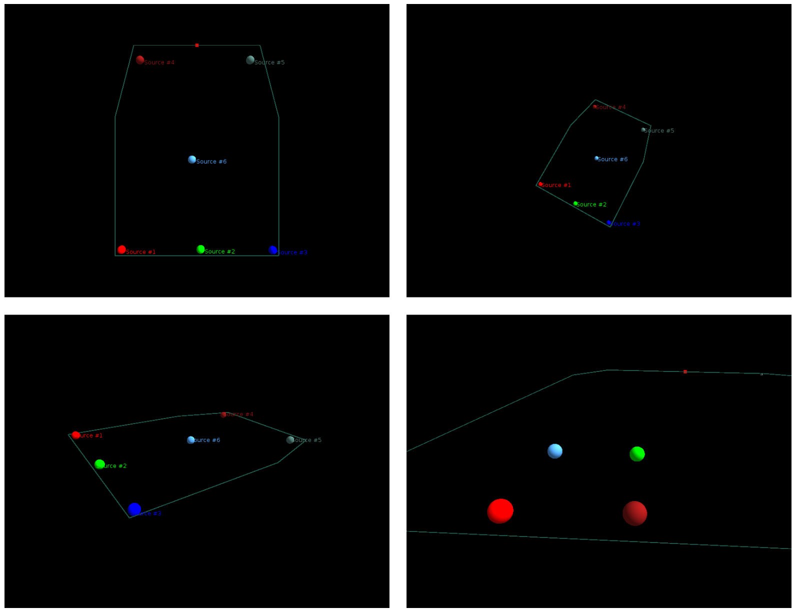


(a) In realtime, we must make use of the anchor points and neighbouring points.

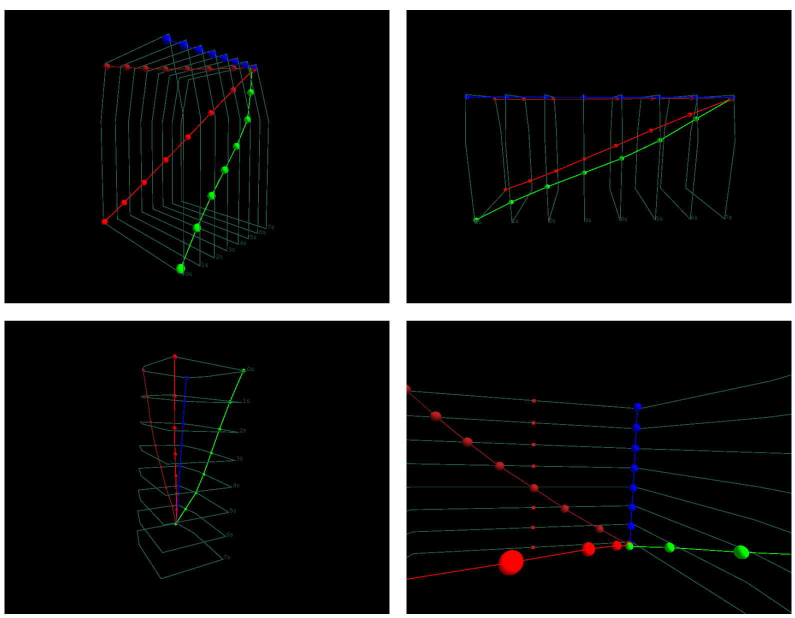


(b) While playing a score, we know the future locations of the source, so we can preload the IRs that correspond precisely to the sound path.

Figure 3.11: Loading grid point impulse responses into cache. The black points are the anchor points and correspond to impulse responses that are always loaded in memory. The red (darkest grey) point indicates the grid point used for the current position, the orange (grey) points the one for which the IRs are currently loaded in memory. The light grey points are the available points on disk.



(a) Scene view



(b) Score view

Figure 3.12: The *xwonder* graphical user interface (pictures taken from [Mon07])

The data is stored in an XML format which is modelled after the XML3DAudio format as described by Potard [Pot06]. In case of very dense data, the score recorder stores the data in a binary file, whose format and path are specified in the XML file.

The format basically has two sections: an orchestra section, which defines which sources are part of the score, and their initial parameters, and a score section, containing all the events of changes in the parameters for each source, as they occur. An example is shown in figure A.5 for a score with 10 sources.

3.7 Graphical User Interface

The graphical user interface, *xwonder* [Mon07], has been written using the Qt-Libraries [Tro05, version 4], and OpenGL. The main functions are created for the realtime use of the system, as well as creating content with the system. The GUI provides an interface to move sources in realtime, to store constellations of source as scenes, and to select scenes. The graphical representation of the sources can be configured by choosing a colour for the source, and giving the source a label. For score manipulation and visualisation, the 3D view offers a representation of the positions of the sources over time, so that the spatio-temporal behaviour of the sources can be seen. Some screenshots of the GUI are shown in figure 3.12.

3.8 Time and synchronisation

There are several concepts of time within the system:

- the user interface can send messages, which have to be executed *now* ($t = 0$) and have a certain *duration*; or it can send messages which have to be executed at a certain time ($t = t_0$) from *now* and have a certain *duration*,
- the score player/recorder has to deal with both MTC and the sample time. It should synchronise itself to MTC, and communicate to the control unit, just like other user interfaces,
- the audio clock.

All communication from the user interface to the control unit about time, is in seconds. As the renderers need to be synchronised with sample accuracy, the control unit translates the time in seconds to frame time. The audio clock is used as the time reference for this. This clock is reliable, has got the desired granularity and is present on each render unit and the control unit. The audio devices in the units are fed with a MADI signal which includes a synchronisation signal[3]. Because the audio links are digital, a sawtooth generated at the control node, will be sufficient to extract the initial synchronisation position from

[3]alternately, the audio devices can also be connected with a Wordclock signal for synchronisation

the audio signal. When initial synchronisation is done, synchronisation will be maintained by the audio sync contained in the MADI signal.

This leads to a system with one central clock and avoids the need for clock skew compensation which is needed when having multiple clocks.

As an example we consider the task of changing the position of a source. This information is sent from the UI to the control unit where a time stamp for this event is generated. Since the control unit has the information about the actual time in samples the messages will be stamped with this time reference and sent to the render unit.

Both the control unit and the render unit can deal with interpolation over time, i.e. it is possible to send the control unit a message to move a source from one position to another with a certain duration of the movement. The control unit will pass on this duration to *twonder*, which then interpolates the movement and calculates the positions (and thus the delays) at the end of each block, and creates the movement. The control unit will also calculate the intermediate positions on the grid, and ensure that the IRs for the intermediate points are preloaded by *fwonder* and the IRs needed for the current position are switched to in time.

3.9 Configuration and data files

For configuration of the system and creating a project with the system, several files are needed to store the relevant data.

It was chosen to use XML[4] for the format for storing this data, as it is easy to extend in case of need.

There are files for:

Default configuration the address of the control unit and working directory (for example: figure A.3).

Array setup positions of the speaker array segments.

Global Array general settings for the array, such as the inner room of the array.

Project the general settings for a project, such as how many sources are used and the characteristics of each sources. It can also contain a reference to a score file, a grid file, and settings for different scenes (static constellations of sources, between which the user can switch). For an example: figure A.4.

Grid the information about the grid points used for early reflection calculation, as well as information about the impulse responses (path and format in which they are stored).

[4]"The Extensible Markup Language (XML) is a general-purpose mark-up language. Its primary purpose is to facilitate the sharing of data across different information systems, particularly via the Internet. It is a simplified subset of the Standard Generalized Markup Language (SGML), and is designed to be relatively human-legible. XML is recommended by the World Wide Web Consortium. It is a fee-free open standard. The W3C recommendation specifies both the lexical grammar, and the requirements for parsing", source: WikiPedia, `http://en.wikipedia.org/wiki/XML`

Score the movements of the sources in time, as described in section 3.6.

In addition to these configuration files, the software package comes with a set of scripts, which automate the starting and stopping of most components.

3.9.1 Array configuration

There are two files for the array configuration: the global array configuration and the array configuration. The first file contains global settings, which all of the renderers need to know, and the graphical user interface. There are settings for:

- The speakers: the distance between the speakers (needed for the calculation for a source moving "through" the speaker array) and the distance to the reference line.
- The calculation of focused sources include: the distance up until which a speaker will contribute to a focused source, as well as the margin for this, i.e. the distance over which it will fade out.
- The elevation compensation: the y-coordinates between which the elevation is, and the (vertical) z-coordinates determining the elevation. This assumes that the coordinate system is chosen in such a way, that the elevation is in the y-direction.
- The polygon, defining the outer limits of the array. This is needed to determine whether or not a speaker should be active, when a source is in front of a speaker segment. If it is inside this polygon, it will be active, provided it is within the "focus"-distance. The graphical user interface needs this information to display the array. In non-closed arrays it should represent the listening space.

An example for the lecture hall H0104 is shown in figure A.1.

The array configuration contains information on the actual speaker setup for the renderer. It contains information on a number of linear segments, where the settings consist of:

- the number of speakers in the segment,
- the start coordinates for the segment,
- the end coordinates for the segment,
- the normal vector of the segment, pointing inwards (towards the listening area),
- a window width, in case a Hanning window is provided as a taper (see subsection 2.6.4).

An example for one renderer of the setup in lecture hall H0104 is shown in figure A.2.

3.10 OSC commands

In table 3.1 an overview is given of the currently working OSC commands. In the next subsections the functionality of these commands is explained, clarifying some of the concepts used in the software.

3.10.1 Project

To be able to store scenes, you need to create a new project with the command: `/WONDER/project/create`, with two *string*s as arguments: the project name and the path.

You can save the project with the command: `/WONDER/project/save`, and later load it again with the command `/WONDER/project/load`.

When a project is loaded *cwonder* will send the calling application the XML representation of the currently loaded project. The calling application needs to be able to understand this message by implementing a callback function to the message `/WONDER/project/current`; the *error no.* will be non-zero if no project is loaded in *cwonder*.

3.10.2 Scenes

You can create a snapshot of the current source positions (called a "scene") and store them in the project, using the command `/WONDER/scene/add` with an integer as argument for the slot number under which you want to store the scene.

Later you can recall the scene with the command `/WONDER/scene/select`, with as arguments the scene number, the time at which the change to the scene should start, and the duration in which it should fade to the new scene.

With `/WONDER/scene/remove` a scene is deleted (and thus the slot is freed again). With `/WONDER/scene/set` you can overwrite an existing scene. Note the subtle difference between adding a scene and setting a scene: adding creates a new scene and stores the current source positions to it. It gives an error back when the scene number already exists. "Set" stores the current source positions to an existing scene and gives an error back if the scene slot does not exist.

With `/WONDER/scene/copy` you can copy one scene to another scene slot.

With `/WONDER/scene/name` you can give a scene a name, a label with which to remember the scene, which is useful for display in the UI.

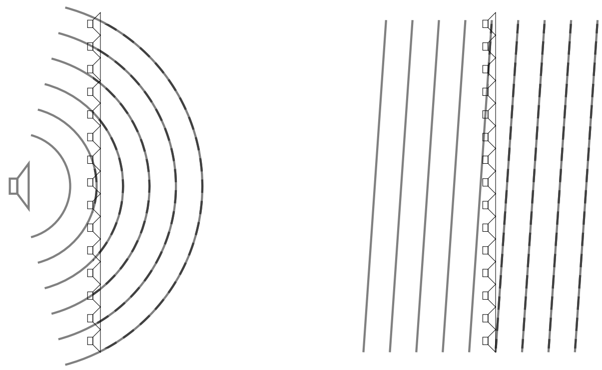
A scene does not need to contain information on all sources: it is possible to enable and disable sources for a scene with the command `/WONDER/scene/source/enable`; it takes as arguments the source id and an integer, which should be "1" when the source is enabled and "0" if it is not. If the source id is "-1" the command enables all sources. With `/WONDER/scene/source/name` a source within a scene is given a label.

3.10.3 Source control

There are currently two types of sources: point source and plane wave (figure 3.13).

command	types	arguments
/WONDER/project/create	ss	project name, path
/WONDER/project/load	ss	project name, path
/WONDER/project/save	ss	project name, path
/WONDER/project/current	is	error no., XML-string
/WONDER/scene/select	iiff	scene no., mix, time, duration
/WONDER/scene/set	i	scene no.
/WONDER/scene/add	i	scene no.
/WONDER/scene/remove	i	scene no.
/WONDER/scene/copy	ii	from id, to id
/WONDER/scene/name	is	id, name
/WONDER/scene/source/name	is	src id, name
/WONDER/scene/source/enable	ii	src id, 1 or 0
/WONDER/global/enable	i	enable
/WONDER/global/fadejump_threshold	i	threshold
/WONDER/global/mute	i	mute
/WONDER/global/max_negdelay	f	maximum negative delay
/WONDER/source/position	ifffff	src id, pos x, pos y, pos z, time, duration
/WONDER/source/angle	ifff	src id, angle, time, duration
/WONDER/source/type	iifffff	src id, type, pos x, pos y, pos z, angle, time
/WONDER/source/fadejump_threshold	ii	src id, threshold
/WONDER/source/mute	ii	src id, mute (1 == silence)
/WONDER/source/enable	ii	src id, enable
/WONDER/source/max_negdelay	if	src id, maximum negative delay
/WONDER/score/quit		
/WONDER/score/undo		
/WONDER/score/redo		
/WONDER/score/play		
/WONDER/score/pause		
/WONDER/score/stop		
/WONDER/score/record		
/WONDER/score/newtime	f	new time
/WONDER/score/reset		
/WONDER/score/offset	f	
/WONDER/score/save		
/WONDER/score/load	s	filename
/WONDER/score/set_midiin_dev	i	midi device
/WONDER/score/set_midiout_dev	i	midi device
/WONDER/score/show_mididevices		
/WONDER/score/status		

Table 3.1: Working OSC commands.



(a) Point source (b) Plane wave

Figure 3.13: Source types.

With the command: `/WONDER/source/type` you can set the type for one source. Plane wave is "0", point source is "1"; the angle argument is the start angle for the plane wave; the position of the source also needs to be set. In the case of a point source, this will be the actual position of the source. In the case of a plane wave, this is a reference point for the calculation; it should be chosen to be a position somewhere behind the array in the direction where the plane wave is coming from. This point determines the basic latency of the plane wave.

`/WONDER/source/position` takes as arguments the source id, the x, y and z position (in meters)[5], the time at which the change should start (in seconds from "now"), and the duration for the change to take place (also in seconds).

`/WONDER/source/angle` takes as arguments the source id, the angle in degrees, the time at which the change should start, and the duration for the change to take place.

`/WONDER/source/fadejump_threshold` takes as argument the source id and the threshold at which the system will stop resampling (causing the Doppler effect) and switch to blockwise crossfades. The threshold is the number of samples which need to be added or removed due to the movement of the source. This is per calculation block, and depends on the calculation block size, the system is configured to.

`/WONDER/source/max_negdelay` takes as argument the source id and the maximum negative delay in seconds, which is related to the maximum distance a source can be in front of the array.

`/WONDER/source/mute` mutes a source, whereas `/WONDER/source/enable` enables the source, so commands about the source are passed on by *cwonder*.

[5]the z position should be 1.0 for now, but may be used in the future

`/WONDER/grain/channel`	iiib	offset time, channel, size, data blob
`/WONDER/grain/point`	ifffib	offset time, pos x, pos y, pos z, size, data blob
`/WONDER/grain/plane`	iffffib	offset time, pos x, pos y, pos z, angle, size, data blob

Table 3.2: OSC commands for spatial granular synthesis. The first command assigns a grain to a specific speaker channel; the other commands to a WFS rendering of the grain.

3.11 Spatial granular synthesis

Within granular synthesis[6] a sound is build up of very fine grains of sound (order of magnitude ca. 1 ms), whose parameters are usually controlled in statistical terms, with parameters for density, frequency range and grain length, with a controllable amount of randomness within each parameter.

If for these parameters also spatial characteristics are set for each grain, clouds of sound can be created which are both spread out through space (these clouds can also move through space), and have an inner spatial structure that yields a new sensation of fine detail within a sonic event, that is not encountered in nature. One could maybe compare it with a swarm of buzzing insects, but then a thousand times denser and not necessarily a buzzing sound.

In order to realise this, it is not feasible to have one audio channel for each grain. Rather an architecture is required, where via OSC messages with the audio data for each grain are sent to the renderers, together with the positional and time information (see table 3.2). In this way, the audio data can be inserted in the delay lines at the required positions for each speaker. As no resampling will be needed, we can process as many grains as required, as the process of calculating the delay offset and amplitude and putting the data in the delay line is a linear operation.

The steps in the calculation are:

1. for each speaker calculate the delay and amplitude for the grain,
2. add the delay to the offset to get the "speaker offset",
3. add the grain samples, multiplied with the amplitude factor, to the samples already present in the delay line of the speaker starting at "speaker offset".

The limits to the density of the grains are thus only related to the network bandwidth and CPU time to process the data.

[6]The theory of granular synthesis was developed by Dennis Gabor, though Iannis Xenakis also has claims on inventing this synthesis approach (Xenakis, Formalized Music, preface xiii). Curtis Roads is often credited as the first person to implement a digital granular synthesis engine. Canadian composer Barry Truax was one of first to implement realtime versions of this synthesis technique. [Wik06]

3.12 Current status of the software

Of the software infrastructure presented in this chapter, not all components have been implemented yet, at the time of writing this thesis. The current TODO list is:

- Inclusion of a single reflection with *twonder*.
- Offline render units (see also chapter 6 for some aspects of this).
- Implementation of the IR caching algorithm in *cwonder*, together with communication between *fwonder* and *cwonder*.
- Score editing operations to the score player/recorder and the GUI. Dynamic anchor point management to the score player.
- The spatial granular synthesis structure needs to be finalised, i.e. an adapted version of *twonder* needs to be created. Some basic structure for distributing the grains has already been done.

Chapter 4

Wave Field Synthesis and electro-acoustic music

4.1 Perspectives

Wave Field Synthesis offers new possibilities for electro-acoustic music, in that it allows more accurate control over movement of sound sources, enables other types of movement than those often seen with circular octaphonic setups (such as movements which use depth and movements through the listening space) and provides an interesting possibility for contrasts between the effect of point sources and of plane waves: a source with a fixed position versus a source which moves along when the listener walks through the sound field. Additionally, it is possible to work with virtual room reflections, even with room dimensions changing over time (see the insert for a thought experiment on relative room dimensions).

The clarity of localisation that is achieved with WFS can also be regarded as a weakness or a challenge for the composer, as Wouter Snoei notes in an email exchange with the author:

> "In regular (non-WFS) spatialisation sometimes the fact that it's not that good benefits the sound. It creates a kind of tension where your ears try to figure out what's happening but never really find out. The spatial position of sounds thus stays unclear, as if a kind of cloak is around it. Obviously partially hidden things can sometimes be much more exciting than the whole thing uncovered. In WFS all spatial positions are always crystal-clear, which can be a bit dull sometimes. It would be nice if one could have the clarity or quality of spatialisation as an extra parameter, and of course having that on a per-sound basis would be a really interesting addition to the power of WFS."

Electro-acoustic music is by its nature more likely to run into the limitations of Wave Field Synthesis, mainly the problem of spatial aliasing (see subsection

Relative room dimensions

From the idea to have a room which changes its dimensions over time, we can start the following thought experiment: imagine a static sound source in a room of which one wall is moving towards the sound source. Then the first sound ray to be reflected from the wall is that which has a path perpendicular to the wall. When the sound ray directly next to that is to be reflected, the wall will already have moved a little closer to the sound source. And so on for the sound ray next to that. So the wall that the sound is actually reflected on, will be a curved wall. Consequently the waves reflected from this curved wall will have a focus point, where all reflected waves will come together again. this curved wall is momentary: for the sound some seconds later the wall will have another position again, and thus the focus point will move as well. Perceptually this could lead to interesting effects.

2.6.4). Due to the distance between the loudspeakers, sounds with a wavelength smaller than this distance, cannot be reproduced accurately by WFS, as spatial aliasing occurs[1]. While in the reproduction of natural sound sources this is usually not a serious problem, as there tends to be a considerable amount of energy in the lower part of the spectrum, electronic sounds can (and often do) have any imaginable spectrum and so the problem of aliasing will be more prominent, when most of the energy is in the upper part of the spectrum.

An important distinction to other applications of WFS is that in electro-acoustic music one does not necessarily want to create a sonic impression that has a natural equivalent, i.e. there is an interest to create sonic events, which would *not* be possible in a natural environment, and only be possible within a synthesised virtual environment, thus creating a new psycho-acoustic challenge for the listener. An example is to create several virtual sound sources, each in another virtual room.

The listening room will influence the resulting acoustic simulation; this is of course also a problem with other loudspeaker spatialisation technologies. Generally it is recommended to use a dry room for WFS reproduction. On the other hand, recent research [WM04, CCW03] has shown that the interaction with the actual listening room will add reflections to the sound, but that these will be the reflections from the virtual source position.

To work with WFS an object oriented approach is needed, i.e. the source material must be divided in such a way, that the materials which will make the same movements through space must be presented on the same channel, and in the production process the material will be linked to data that describes the spatial transformations of the material. For many composers this way of working will not pose a large problem, as conceptually many of them are already using this approach when working with spatialisation. The only difference will be in the production process, where they no longer have to render to a fixed

[1] For a speaker distance of 10cm the aliasing frequency is 1.7 kHz in the worst case, 3.4 kHz in the best case, for a speaker distance of 15 cm. to 1.1 kHz and 2.2 kHz respectively.

number of channels, which will then constitute the piece to be performed.

4.1.1 Recording techniques

Recording techniques for Wave Field Synthesis can be distinguished based on the effect that is aimed at, namely recording

1. a single sound, which will later be spatialised,
2. the acoustics of a space, which will later be used to add these acoustics to virtual sound sources,
3. a real environment, which will be reproduced as is, i.e. there will be no further adaptation of the spatialisation in the production.

For the first, it is important that the source signal is as dry as possible, so the recording should only contain the direct sound. The source can then be reproduced as a point source with WFS.

There are also methods to record the directivity pattern of a sound source for use with WFS reproduction [JAMdV05].

To record the acoustics of a space microphone arrays need to be used. There are several techniques to process the measurements for use on any WFS configuration, for example a plane wave decomposition [HdV01]. Theoretically, such measurements need to be made for each source position that the virtual source will take within the room, however in practice we can measure the responses for fewer source positions, as we do not hear small differences in reflection patterns [Hel03]. The results of these measurements and processing can then be used to add virtual room reflections to virtual sources.

Measured room impulse responses can be further analysed and processed to allow for artistic control over the reflection patterns of the virtual room. Tools for this are being developed by Melchior [MdV06, dVLM06], and a screenshot of the user interface is shown in figure 4.1. The idea is that after analysis of a recorded wave field, several mirror image sources can be identified, which contribute to important early reflections. After having identified those, they can be moved, to create a different pattern. Furthermore, the other reflections are split up based on directions from which they come, through a plane wave decomposition. The spectra of these can then also be edited to create the desired result.

Recording a real environment for reproduction by WFS is of interest for compositions based on field recordings. Here also microphone arrays will be needed to record the spatial properties of the environment, and the same post processing techniques can be used. However, whereas microphone array measurements for room impulse responses are usually made by moving the microphone after each measurement, for recording a real environment an array is needed where all channels are recorded simultaneously [HSdVB03, AdV04, dVHG07]. Such an array has been developed at the TU Delft (see figure 4.2), and the prototype has already been used to record a new music performance, which uses space as a compositional parameter.

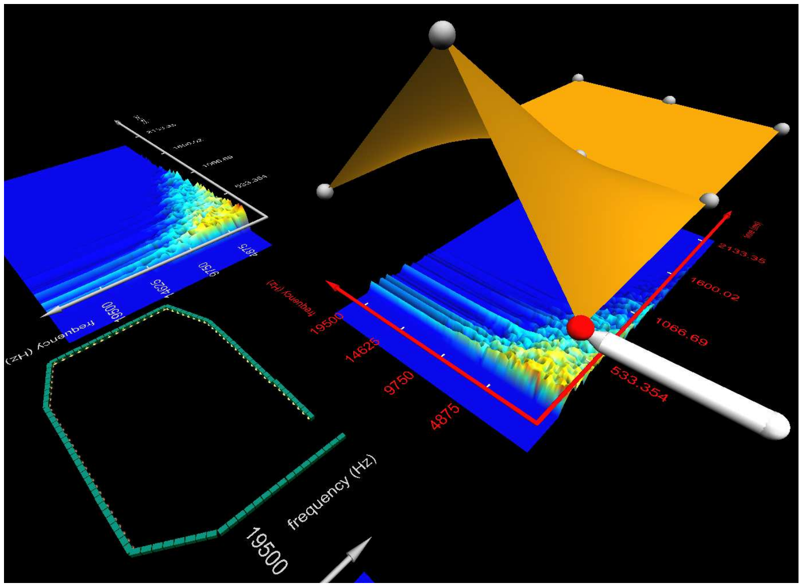


Figure 4.1: A screenshot of the user interface for editing reflection patterns of a virtual room (from [MdV06]).

Figure 4.2: A circular recording array, as developed at the TU Delft [HSdVB03]. Here in action for a recording in the Liebfrauenkirche in Halberstadt, Germany (*Photo by Bernhardt Albrecht*).

4.1.2 Sound installations

As the wave field created with WFS is stable in the sense that sources stay where they are when walking around as a listener[2], this is an interesting technology for use in sound installations. Within a sound installation the listener is usually not bound to one listening position, as with seated concerts. Thus, the listener can become an active participant in the installation, by moving around within the sound field, and thus gaining several perspectives on the piece (see also chapter 1 for a discussion on this topic).

Note, this difference of perspective on the sound field, based on the listener's position, is also an important issue for concert situations. It has been argued that with WFS there is no sweet spot, but rather a large listening area. However, this does not mean that everyone will have the *same* listening experience[3], instead it is comparable to having different seats during a concert of an orchestra, where similarly the impression of the acoustic event depends on where one is seated in the concert.

4.1.3 Realtime use

With the current tools available it is possible to change the positions of sound sources in realtime, and thus make this dependent of a real time process in an installation. This was done for example in the installation *Scratch* (2004), described below.

Latency of the system may be a concern, though depending on the actual hardware setup, latency can be minimised: the WFS processing of the TU Berlin system introduces a latency of 256 samples[4], i.e. 5.8 ms at 44.1 kHz; this was found sufficient for Tutschku's *Zellen-Linien*, where the live sound of a grand piano was processed and played back via the WFS system. This number is without the latency introduced by the sound travelling from its virtual position to the listener. In the case of focused sources, there needs to be a pre-delay, which needs to be configured beforehand.

If sound is recorded, processed and send back over the system, feedback may become a problem, as not all traditional methods to prevent feedback may be applied, due to the large amount of speaker channels involved. This is still a topic for future research.

4.2 Available systems and tools

Besides the system at the TU Berlin, there are several other WFS systems available to composers nowadays, and an overview will be given in this section.

[2]as opposed to spatialisation with stereophonic techniques where the localisation of a source tends to move along with you as you move out of the sweet spot, until it collapses onto the speaker.

[3]listening experience in a narrow sense, i.e. resulting from the same perceivable acoustic event, and not taking into account extra-acoustical parameters influencing the experience.

[4]in fact, this latency is dependent on the block size and can be configured. Larger block sizes are computationally more efficient and allow for a more stable system with more sound source inputs.

Figure 4.3: WFS speakers for the *The Game of Life* system (August 2007)

4.2.1 *The Game of Life* - WFS SYSTEM

In 2006, a mobile Wave Field Synthesis system [Sno06] was realised for the Dutch foundation *The Game of Life*. It consists of a loudspeaker array (designed by Raviv Ganchrow) of 192 SEAS coaxial 2-way loudspeakers, 12 active subwoofers (Hypex) (see figure 4.3), driven by two G5 computers running software written in SuperCollider [McC] by Wouter Snoei and Jan Trutzschler. The system was premièred at November Music 2006[5] in Gent (BE) and Den Bosch (NL). Composers Barbara Ellison, Yannis Kyriakides and Wouter Snoei created pieces for this system. For the SuperCollider Symposium[6] in September 2007 in The Hague, several more composers created pieces for the system. Further projects utilising this system are being planned, such as the interdisciplinary project *Sternenrest* about science, art and contemplation [Ste06], with composer Wim Bogerman. The system is currently installed in the Scheltema Complex in Leiden.

The Institute for Sonology is in close cooperation with the Game of Life foundation and one of the topics of the "sound and space" course[7] is wave field synthesis.

The user interface for this system has been created in SuperCollider on MacOSX, and provides many useful graphical tools for spatial composition. The interface provides a score editor, where groups of audio files and paths can be arranged. A single group can then be edited in more detail, and single paths can be edited using the path editor as shown in figure 4.4. The strengths of this

[5] `http://www.novembermusic.net`
[6] `http://supercollidersymposium.sampleandhold.org`
[7] `http://www.koncon.nl/public_site/220/Sononieuw/UK/frameset-uk.html`

interface are that default forms are provided (such as circles, spirals, random motion, etc.), points can easily be added, deleted and moved in the path, and the path as a whole can be scaled, rotated, duplicated, and so on; it is also possible to store the path as a SVG (scaled vector graphic) file, and edit it with a graphics program like Adobe Illustrator[8] or Inkscape[9]. Especially interesting is the view at the bottom, where the speeds of the source to the next point are shown; this will give an indication of the Doppler effects that will be heard, and the view provides ways to edit the path in time on this level (e.g. equal times between points, or equal speeds), or simply by dragging the points up and down for faster and slower speeds. When a path has been defined, it can be stored in a bank, for future use, or be assigned to a selected sound file.

4.2.2 Casa del Suono

In the Casa del Suono[10] in Parma, Italy, two systems are installed that utilise the WFS principle. One is a listening room, equipped with WFS, the second is a sound chandelier, the *lampadario* (see figure 4.5), used to focus sound sources.

For the inauguration of the exhibit, Martino Traversa created a composition titled *NGC 353*, from the name of a spiral galaxy located 8 millions light years from Earth. The composition is a sort of "sonic sculpture", dancing around and above the head of the listener, who gets the feeling of being able to touch the sounds.

4.2.3 Commercial systems

To date there are two companies that market complete WFS systems, *sonicEmotion*[11] and *IOSONO*[12], including hardware and software to do WFS.

The Fraunhofer Institute for Digital Media and IOSONO have installed larger systems at the Lindenkino in Ilmenau, in Bregenz [Rod05] and in Bavaria Filmstadt. *IOSONO* provides software for rendering and control of the sources, and have worked together with the company *LAWO*[13], to integrate their software interface with a mixing desk.

The Swiss company *sonicEmotion* are in close cooperation with IRCAM in Paris. Their software is controllable via OSC, and they created an interface to IRCAM's *spatialisateur (spat)*, as well as a plugin for the *Pyramix*[14] multi-track software.

Other companies are starting to market Wave Field Synthesis as well, like ACS[15] [dVvDSvdH07].

[8] `http://www.adobe.com/products/illustrator/`
[9] `www.inkscape.org`
[10] see `http://pcfarina.eng.unipr.it/CdS/CdS.htm` and `http://www.danieletorelli.net/lampadario.html`.
[11] `http://www.sonicemotion.com`
[12] `http://www.iosono-sound.com/`
[13] `http://www.lawo.de`
[14] `http://www.merging.com/`
[15] `http://www.acs-bv.nl/`

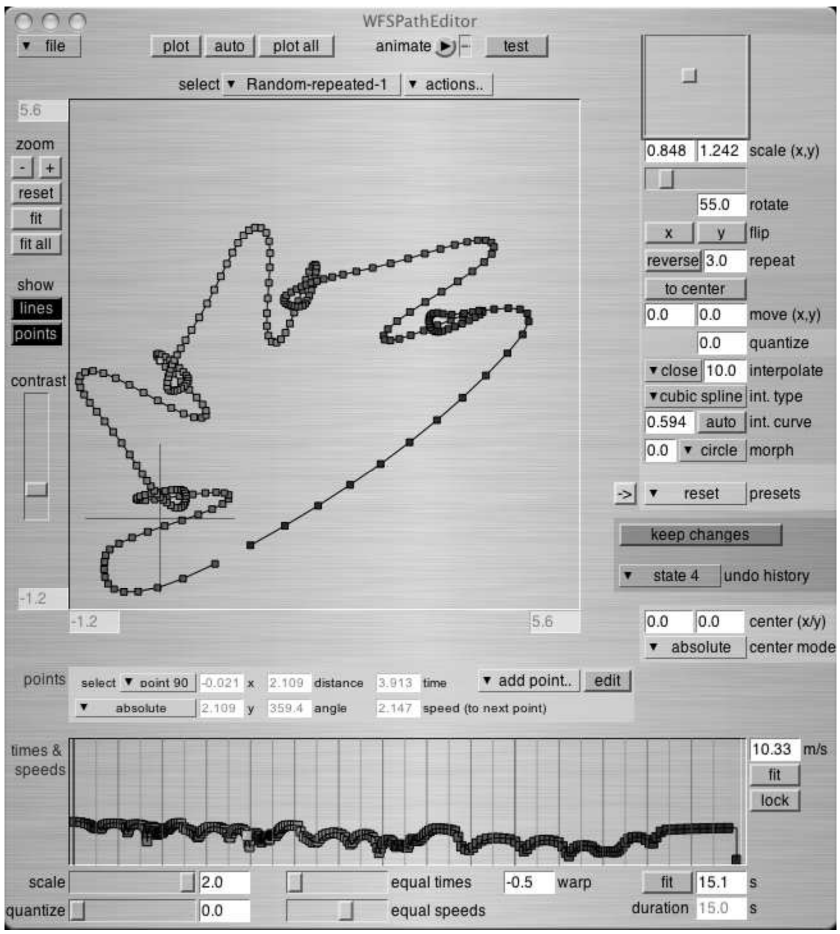


Figure 4.4: The path editor of the WFS software by Wouter Snoei.

Figure 4.5: The Lampadario in Parma, Italy (Photo taken from `http://www.danieletorelli.net/lampadario.html`).

4.2.4 Further research systems

Several research institutes have invested in WFS systems, these institutes include the sound control group of the TU Delft[16], IRCAM[17], the Institute for Telecommunications and Multimedia Applications (iTEAM) of the Technical University of Valencia in Spain[18], Fraunhofer Institute for Digital Media[19], Institut für Rundfunktechnik (IRT)[20], the University of Erlangen[21], Fachhochschule Düsseldorf[22], the T-Labs of the Deutsche Telekom[23], Digitale Medien of the Furtwangen University[24], McGill University, Montréal, Canada[25] and University of California[26], Santa Barbara, USA.

Most of these institutes are interested in collaborations with artists, though this is not the primary focus for all of them, e.g. the University of Erlangen presented on the Hörkunstfestival 2005 in Erlangen, Germany, several works[27]

[16] `http://www.soundcontrol.tudelft.nl`
[17] `http://http://recherche.ircam.fr/equipes/salles/WFS_WEBSITE/Index_wfs_site.htm`
[18] `http://www.gtac.upv.es`
[19] `http://www.idmt.fraunhofer.de/`
[20] `http://www.hauptmikrofon.de/`
[21] `http://www-nt.e-technik.uni-erlangen.de/lms/research/projects/WFS/`
[22] `http://www.medien.fh-duesseldorf.de/`
[23] `http://www.deutsche-telekom-laboratories.de/`
[24] `http://www.dm.hs-furtwangen.de/`
[25] `http://www.cirmmt.mcgill.ca/`
[26] `http://www.create.ucsb.edu/`
[27] `http://www.hoerkunst.de/hkf2005_hkf3-B10.html`

of various composers.

From the author's experiences with users of the *sWONDER*-software, there must be at least a dozen or so other small WFS systems, set up by artists (e.g. McMahon as described below) and by researchers in small companies.

4.2.5 Compatibility between systems

There have been a number of different systems created, both differing in software implementation, as well as hardware setup (number of speakers, array form, etc.). With all these different systems, the question arises whether compositions are transferable to other systems. This is still a problem that needs to be solved, and currently efforts are being undertaken to agree upon a common file format to store the composition data.

Though there are existing standards, such as MPEG4 [PBG03], which include spatial audio coding, these standards do not cover all rendering possibilities that WFS is capable of; also the use of compressed audio is not desirable for electro-acoustic music. Furthermore, it is not an free standard, which makes implementation in open source software problematic. Another existing format is XML3DAudio [Pot06], which could serve as a basis for a common format.

The format should (for each virtual source) contain information on

- position,
- type (plane wave, point source, directional source, 3D object),
- room effects

for each moment within the composition. Additionally, it is of interest to be able to store scenes: i.e. fixed constellations of virtual sources (see also chapter 3). Details of the speaker array may be useful to know for which system the composition was originally created. A point of reference in the coordinate system should also be defined.

There are still a number of problems to solve to make a compatible format: as the software implementation details differ between systems, and it is unclear what effects these have on the audible result, e.g. some systems include Doppler effects within the source movement, or have this as an option, whereas other systems do not; another example is the multichannel equalisation for the flat panel loudspeakers [CHP02, Cor06]. Not all implementation details are well documented, or the documentation (or the source code) is not open, which poses another complication.

Furthermore, it is still questionable how well compositions made for small systems (and thus usually also a smaller space) scale up to large systems, and vice versa. It is unlikely that there will be a straightforward way of scaling compositions: a composer may have audio material on different tracks that are related to each other (together forming a sound object), which if they are scaled up like they were single tracks may loose their common sound impression. When working with a larger system, it may make sense to have more separate sound sources[28], as more sound sources will be distinguishable by the listener, whereas when downscaling, several tracks could be combined. An issue here is also the question of intimacy, which was discussed in chapter 1; a piece composed for a small room tends to be much more intimate than a piece for a large room. There may be cases where this change in scale just does not make sense.

Compatibility with other spatialisation technologies is also an issue. Each spatialisation technology has its own advantages and limitations, and it depends on the techniques used with the technology, whether or not the same effects can be reached with another technology. Within these limits, it is possible to achieve compatibility with ambisonic and binaural formats. This compatibility has as an advantage that composers can prepare their work on such systems, before having access to the actual WFS system. On the other hand: both in ambisonics and binaural techniques, it is possible to add the vertical dimension, which up to now, has not been implemented yet for WFS, due to technical and financial reasons (see also chapter 2 for a short discussion on this).

4.2.6 Economical issues

In the past years WFS has gained acceptance as a feasible alternative to other surround sound systems, within audio engineering circles[29]; nonetheless, the market does not seem ripe yet for this technology.

One of the issues is that there is of course not much content yet available, which really utilises the possibilities of WFS. Partly, this is a deadlock, as entertainment industry will not produce content for a medium that is not widely spread. This seems to be one of the main problems for marketing WFS to the film industry, though an additional factor there is also that Dolby Surround has only recently become a market standard and the majority of the theatres still have a long time before they can invest in new audio technology.

However, there are composers and other artists interested in exploring the possibilities of WFS, and it is just a matter of giving these people access to the technology. Doing so, will not only provide content for WFS, but also give impulses for new demands on WFS systems; only by working with the system, desires and ideas for tools will surface, and drive new developments and research. Getting composers involved in the development also has an economic advantage: as they create content for the reproduction method, there will be a demand for more reproduction spaces.

A market as of now neglected by the manufacturers is the club scene. Here there is already an establishment of surrounding the dancing audience with speakers, but mostly these speakers are not used for spatialising the music, but are meant to enhance the overall sound level of the system. Composers and musicians of club music are not uninterested in spatialisation and it is the author's belief that by involving some of these composers to create music with WFS,

[28] This was the case with *Rituale*, Tutschku chose to use twice as much virtual sound sources when adapting the piece for the TU Berlin system.

[29] Diemer de Vries noted this in a conversation with the author during the AES Conference 2007 in Vienna.

and experimenting in the club environment with this technology, could become a great success. Moreover, the club scene is less bound to industry standards than film industry, and often compete with each other by special features of the club, ranging from providing a certain ambience, certain performers and DJ's, music style/genre, and possibly advanced sound systems.

Luckily, in the recent years WFS is gaining a lot of interest from educational institutes, and slowly they are setting up WFS systems and teaching students about the technology, as well as giving them access to work with it. Obviously, this is an important way to ensure that WFS will become a market potential.

The compatibility issues of course need to be looked into and solved, if WFS is to become a standard for the mass market.

For WFS systems at home there are certainly issues to be solved on how to place speakers inside a living room environment. This is one of the main reasons there is an interest in the flat panel loudspeakers, as these can be more easily integrated with the living room architecture.

As loudspeaker companies are becoming more interested in developing and marketing multichannel loudspeaker panels, the cost for the loudspeakers can be expected to decrease, so this will be less of an impeding factor.

4.3 Compositions and installations

Edwin van der Heide's sound installations

Edwin van der Heide[30] has created several sound installations which use the principle of sound holography, by recording a sound field with an array of microphones and reproducing it via a similar loudspeaker array, such as "*A World Beyond the Loudspeaker* (1998): an installation with 40 channel wave field recording and wave field synthesis", which utilises a planar array of speakers. The sound material for this piece was a recording with the microphone array near a rail road track, while a train is passing. In "*Wavescape* (2001): underwater wave field recording with acoustic wave field synthesis", he uses a linear array of hydrophones, which record the underwater sound, which is then played back directly via much closer spaced speakers on the waterside; the ratio between the speaker distance and the hydrophone distance correspond to the ratio of the sound velocity in air and in water. In these two works, there was a one-to-one relationship between the microphone recordings and the loudspeaker reproduction.

Impuls #6 (2000) was created for the same plane of 40 loudspeakers as *A World Beyond the Loudspeaker*, but here the necessary delays and amplitude factors for the (virtual) sound sources and reflections were calculated with a MAX/MSP [Max] patch, thus the wave field was synthesised.

[30] `http://www.evdh.net`

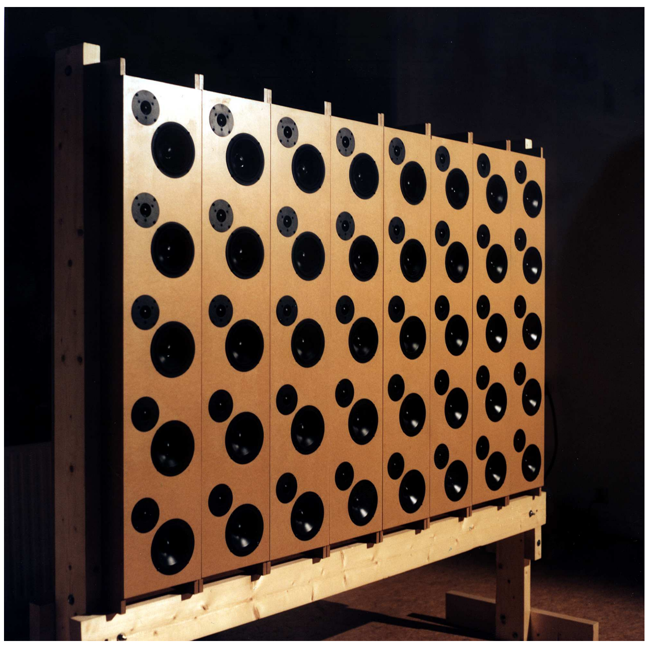


Figure 4.6: The loudspeaker array of the sound installations *A World Beyond the Loudspeaker* (1998) and *Impuls #6* (2000) by Edwin van der Heide.

Artist	Title	year
Marije Baalman	Pollocks Sprechwunsch	2003
Frieder Butzmann	Bunte Flügel	1974/2003
Boris Hegenbart	hermetische_garage	2003
Marc Lingk	Pingpong Ballet	2003
Robert Lippok	Hands and Fingers	2003
Markus Schneider	Ballroom	2003
Ilka Theurich	Restored to Life	2003
Marije Baalman	Beurskrach	2003
Marije Baalman	Scratch	2004
Kirsten Reese	Hallenfelder	2006
Hans Tutschku	Rituale	2004/2007
André Bartetzki	Reale Existenz!	2007
Christian Calon	East (from Atlas)	2007
Victor Lazzarini	Streams	2007
Ludger Brümmer	Xronos	2005/2007
Michael Amman	Oral 29	2007
Shintaro Imai	Immersive Motion Study	2007
Hans Tutschku	Zellen-Linien	2007

Table 4.1: List of WFS compositions created in Berlin from 2002 to 2007. Program notes can be found in appendix B.

4.3.1 Works from the Electronic Studio of the TU Berlin

Several composers have created compositions for the Wave Field Synthesis systems at the TU Berlin (see table 4.1 for a complete list), which explore the artistic possibilities of Wave Field Synthesis.

The system used for the first compositions (2003-4) was a frontal linear WFS array consisting of 24 FOSTEX speaker with a distance of 12.5 cm (see section 3.1). The compositions from 2003 were created for and presented at the festivals Club Transmediale (Berlin (DE), February 2003) and Electrofringe (Newcastle (AU), October 2003). At the Linux Audio Conference (Karlsruhe (DE), April 2004) the sound installation *Scratch* was presented on this same system.

For the Musikfestival Donaueschingen (DE, October 2006), the installation *Hallenfelder* was created using a frontal linear array of 20 loudspeakers with a distance of 35 cm, thus spanning at total of 7m.

In 2007, the WFS system for the lecture hall H0104 (see section 3.1) was completed and several more compositions were created for the Linux Audio Conference (Berlin (DE), March 2007), the official opening of the hall and the Lange Nacht der Wissenschaften (both June 2007).

The software with which these compositions were created has been described in section 3.2 and in [Baa03, BP04, Baa05] previous versions of the software have been discussed.

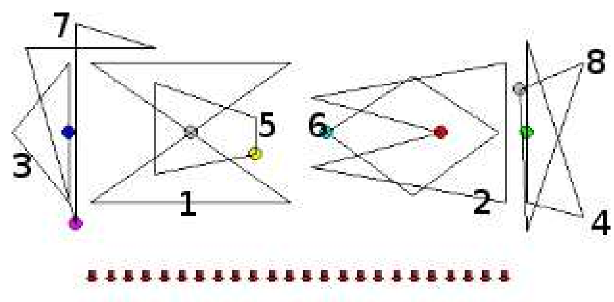


Figure 4.7: Sound source movements in *Ping Pong Ballet* of Marc Lingk. Paths 1 & 2 are the paths of the ball bouncing on the table, 3 & 4 of the ball being hit with the bat, 5 & 6 of multiple balls bouncing on the table, 7 & 8 of balls dropping to the floor.

Ping Pong Ballet (2003) - Marc Lingk

The sounds for this piece were all made from ping pong ball sounds, which were processed by various algorithms, alienating the sound from its original. Using these sounds as a basis, the inspiration for the movements was relatively easy as the ping pong ball game provides a good basis for the distribution in space of the sounds. Thus, Marc Lingk created various loops of movement for the various sounds as depicted in figure 4.7. Choosing mostly prime numbers for the loop times, the positions are constantly changing in relative distance to each other. The movement is relatively fast (loop times are between 5 and 19 seconds). In the beginning, the piece gives the impression of a ping pong ball game, but as it progresses the sounds become more and more dense, creating a clear and vivid spatial sound image.

This piece explores the boundaries between a defined perceptually localised sound event (a single ball hitting the table on a specific location) and spatially displacing dense event structures to enhance the overall liveliness of the sonic impression, and the transition between these two extremes. This transition can be seen as an example of the temporal development Barrett mentions [Bar02] (see also chapter 1): through the associations a listener has with the sounds in the beginning of the piece, the spatial imagery in the later part becomes that of a surreal ping pong ball game where first the number of balls increases enormously, after which they become like a bubbling fluid.

Pollock's Sprechwunsch (2003) - Marije Baalman

In this piece the movements are based on a painting created before in a rather improvisational way. The different colours in the painting were mapped to different sounds and they also have different movement characteristics. One source is moving perpendicular to the array, another parallel to the array. Yet another is zigzagging to and from the array, one source is jumping from one location to another. The other two sources have other types of paths that are less easily stereotyped; in figure 4.8 an overview is shown of all the movements. The exact movement in time is made dependent on the sounds. Silences on the sound input are used to let the virtual source jump to another position for the next sound to start its path. As the movements are relatively slow and the sound is not very dense, the movements and different positions of the sound can be heard quite clearly. Through the different spatial relationships between sounds, depth effects also become perceptible. The space is being defined by the sounds, their locations and their choreography.

The sound material itself consists of recordings of sounds created with balloons: bouncing, squeaking, rubbing, exploding, rolling, inflating and deflating. This basic material has been processed with SuperCollider [McC] with vowel-like filters, which give the material a speech-like character, without becoming actual speech ("Sprechwunsch" is German for "the wish to speak").

Restored to Life (2003) - Ilka Theurich

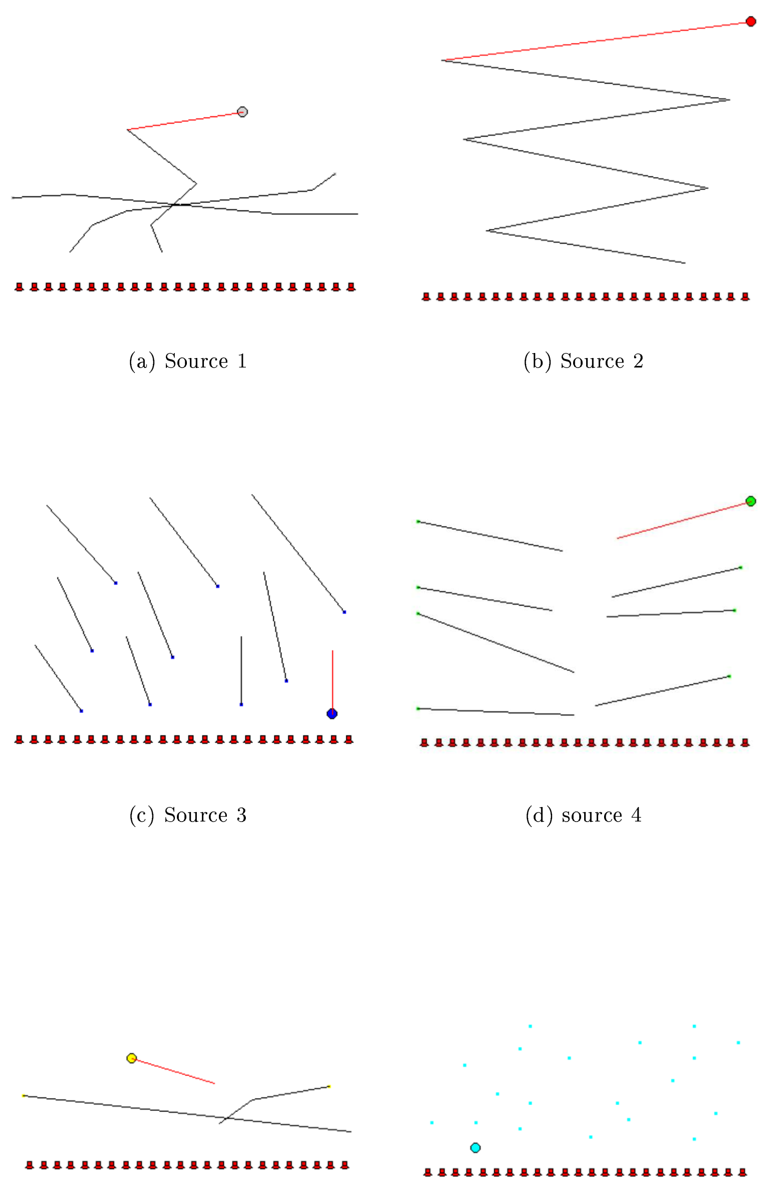
Ilka Theurich, a student of sound sculpture in Hanover (in 2003), was mainly interested in the possibilities of including virtual rooms and reflections into the composition. In her piece, one sound is placed in a rather small room with fully reflecting walls (figure 4.9). This results in a sound that was virtually at several locations (due to the mirror image source model). As the sound from the actual source location is the first sound to reach the listeners' ear, the sound is still located there by the listener.

Other sounds are placed in a larger room, while others are moving without being placed in a room. One of the sources is a plane wave, which allows the listener to get different perspectives on the composition by moving through the listener area. The plane wave sound only has a direction and as such is always in front of one, with a specific angle, whereas the other sounds have clearly defined locations.

The effect of the movement and reflections proved to be the most clear for recorded sounds (having a rich spectrum), as opposed to synthetic sine-based tones. In order to limit the CPU-load, some compromises had to be made: the total amount of reflections calculated was reduced.

During this work (beginning of 2003) the idea came up to enable the room characteristics to change in time. This has been implemented in the software since then, though the idea has also led to the thought experiment described on page 72.

The work also shows that in these compositions room reflections are not necessarily used to simulate a realistic room, but rather to give the sound an



(a) Source 1 (b) Source 2

(c) Source 3 (d) source 4

(e) Source 5 (f) Source 6

Figure 4.8: Source movements in *Pollock's Sprechwunsch*.

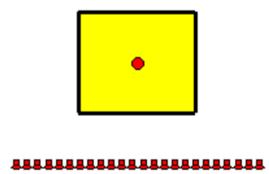


Figure 4.9: A sound in a very small room in *Restored to Life*; the room is smaller than (and outside of) the actual listening area.

extra characteristic, which with other technologies would not have been possible to achieve. This can be compared with the use of reverb not as a means to add a room expression, but to create a special effect, after which the reverberated sound can be seen as new material by itself.

Spatial trajectories of pure synthetic sounds were not very succeful in this piece, as they were perceptually hard to localise.

Beurskrach (2003) - Marije Baalman

In the composition *Beurskrach* four sources are defined, but regarded as being points on one virtual object[31], i.e. these points make a common movement; the sound material for these four points are also based on the same source material, but slightly different filterings of this, to simulate a real object where from different parts of the object different filterings of the sound are radiated. These slight variations in the filtering were often perceived as phasing effects, which suggests that the spatial resolution of the object was not high enough (see also chapter 7).

During the composition, the object, a shopping cart, comes closer from afar and even comes in front of the loudspeakers, there it implodes and then immediately scatters out again, making a rotating movement behind the speakers, before falling apart in the end. See figure 4.10 for a graphical overview of this movement. The composition plays with exaggeration of the natural spatial perspective: as the object is far away the perspective is emphasized by placing the source points closer together and as the object approaches, the source points move further apart from each other.

The piece was accompanied by a video created by Julius Stahl, using images of shopping carts.

[31] This piece can be seen as the author's first artistic exploration for the use of displaying larger sources with Wave Field Synthesis

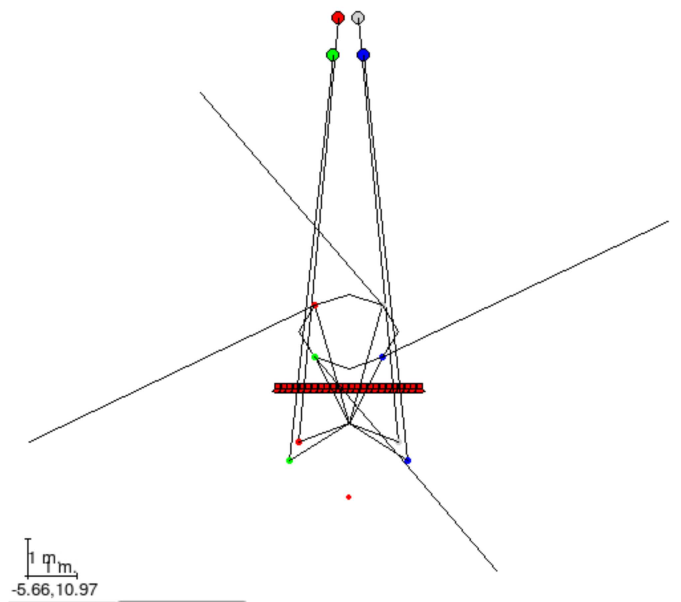


Figure 4.10: The sound source movement in *Beurskrach*.

Scratch (2004) - Marije Baalman

The sound installation *Scratch*, that was presented during the Linux Audio Conference 2004, makes use of the OSC-control over the movements. The sound installation is created with SuperCollider, which makes the sound and which sends commands for the movement to WONDER in a generative, interactive process. The concept of the sound installation is to create a kind of sonic creature, that moves around in the space. Depending on internal impulses and on external impulses from the visitor (measured with sensors), the creature develops itself, and makes different kinds of sounds, depending on its current state. The name "Scratch" was chosen because of two things: as the attempt to create such model for a virtual creature was the first one, it was still a kind of scratch for working on this concept. The other reason is the type of sound, which sound like scratching on some surface.

The sound is actually built up out of two components; one being a low frequency breathing sound, the other being the scratching sound. Within the composition the breathing sound is enhanced by being placed in a virtual room, and it proved better to limit this part of the sound in its movement to a smaller area than the scratching sound. The breathing sound created undesirable artefacts when allowed to make larger movements, as the FIR coefficients were crossfaded; the low sound was also hard to localise perceptually. The scratching sound on the other hand was very well localisable and its movement was clearly audible. Space is used in this piece for creating spatial trajectories of a single

Figure 4.11: The sound installation *Scratch* during the Linux Audio Conference. In the ball are accelerometers to measure the movement of the ball, which influences the sound installation (*photo by Frank Neumann*).

sound object, while the listener can move freely through the listening area, and influence the sonic movement by interacting with the installation.

Hallenfelder (2006) - Kirsten Reese

The sound installation *Hallenfelder* was created for the *Donaueschinger Musiktage* and used recorded sounds as its basic material. The recordings were made in nine halls in Donaueschingen, used as concert halls during the *Musiktage*, but which normally host other activities, such as sports, Carnival parties, flea markets, exams, or cinema. On the one hand the basic room noise, caused by ventilation and heating and other noises which penetrate and resonate in the empty hall, was recorded, and on the other hand specific events taking place in the halls. Certain sounds were also recorded with a mono microphone to get just the direct sound in the recording.

The main challenge during the production of this sound installation was to create a spatial impression of the rooms with the recordings made, though the aim was not so much to create a completely natural reproduction of the room sounds, but in a way accentuate it. The four channels of the recordings were positioned as point sources within the room, in an approach similar to virtual panning spots [TWR02], i.e. the points were used as virtual loudspeakers between which the sound is panned, or in this case they correspond directly to the microphone positions and thus reflect the spatiality in the recording. For different scenes the actual locations of the point sources were different and transitions from one scene to another were achieved by moving the point sources to their new location during a certain time span. For some isolated events, where there were recordings of the direct sound available, a moving point source was used to move the sound through the space; partly these were sounds that you expect to move, such as a basketball and footsteps, but this was also used for sounds that normally do not move, such as the sound of someone jumping on a trampoline and the opening of a snack package. This effect was used sparsely, but effective.

Figure 4.12: The loudspeaker array in the sound installation *Hallenfelder* in the small sport hall of the Realschule in Donaueschingen.

The result was very good: due to the wave field synthesis, a very spatial feeling was achieved, and during scenes where there was a lot of activity, the impression was very realistic: one could really imagine the people making the sounds in the sport hall, without having the feeling that the sound was coming from the loudspeakers. Despite the relatively large distance between the speakers, there were no perceptually disturbing aliasing effects.

The work clearly tries to recreate existing spatial sound fields, but at the same time creates illusions of moving sounds, thereby alienating the natural sound field. Different spatial sound fields are shifted one into the other, moving the listener from one space to another. The video helps the listener in recognising the sonic environment, by giving short visual excerpts of the activities in the hall, while at other times showing an image of the current hall where it is empty. The documentary quality of the piece has surely contributed to the great interest in the work from the local population, while for the regular visitors of the Musikfestival the piece gives an insight into the everyday lives of the nine "concert" halls documented.

Rituale (2004/7) - Hans Tutschku

"The 15 minute piece *Rituale* (2004) processes human voices and instrumental sounds from various cultures to a sound ritual. It is a continuation of Tutschku's work on *Rojo* and *object-obstacle*, which were both also concerned with the theme of rituals. The composition uses extensively the possibility to place sound sources close to the listeners, i.e. inside the listening room."[32]

Rituale was originally (in 2004) created for the IOSONO Wave Field Synthesis system in the Ilmenauer Lindenkino, and was adapted for the lecture hall of the TU Berlin in 2007. During the adaptation, Tutschku chose to split up the sound material in twice as many separate sound sources, as due to the larger space in which the sources are placed, the listener can distinguish between more sources simultaneously. For the control of the movements of the sound, the composer designed his own user interface in Max/MSP [Max], sending OSC-messages to *sWONDER*. The resulting score was recorded as well with *sWONDER*'s score recorder, so that the piece can be played back without the need for the Max patch.

The piece mainly explores the use of spatial trajectories of various sounds and intrusion of the listener's intimate sphere by moving sounds very close to the listener.

East (from Atlas) (2007) - Christian Calon

East is one out of four parts of a concert installation by Christian Calon, and has been produced for the Berlin WFS lecture hall. The composition was prepared as 8 channel audio material, which was then spatialised on the system, in two groups of four channels. For this Calon used the LADSPA plugin (see section 3.2) to control four related sources, which move together through the room.

The four channels in each group are like a square, and he mostly varies the front-back position, moving the sources through the room, while the left-right position has only few variations. Also the scaling, i.e. the distance between the four sources is changed in both dimensions, and later in the piece there are also rotations of the four sources around the centre point.

The main stage in the piece is in the front centre[33], from where the sound is drawn into the space, creating an immersive effect, where the listener is inside the sound scape. Throughout the piece, he keeps returning to the front centre, as an anchor point. Calon seems mostly interested in a spatial diffusion of the sound, rather than clear spatial trajectories, immersing the listener in the acoustic realm of his sound material.

Reale Existenz! (2007) - André Bartetzki

Bartetzki takes a similar approach, in that he also uses groups of four sources (again making use of the LADSPA plugin in Ardour). But he also uses some single sources for single actors, like a voice (Schroedinger), and a squeaking

[32] see the program notes in the appendix B.
[33] coincidentally, the front of the hall is towards the east

sound that moves through the room. In addition he uses plane waves for distant metallic sounds. In total he uses 32 audio streams, divided into 8 plane waves (static), 4 single sources, and 5 groups of 4 sources. In this way he creates various overlapping sonic spaces, where single sound objects have very clearly defined spatial trajectories.

Streams (2007) - Victor Lazzarini

In the piece *Streams*, sounds of recognisable sources are split apart in space and timbre; so space is tied in with spectra-morphology. The piece starts with entries of the four woodwind instruments in different sides of the room, then their spectra are split in two and glide the components in opposite directions as they move apart in space. Eventually they meet together 180 degrees of where they start, fusing into their original spectra. In other sections of the piece, the instruments' spectra are split in four and then move around the room, spinning, etc. So he uses spatial trajectories for his source material, but breaks up their natural coherence by splitting up the spectra. Thus it becomes a challenge for the listener as his ability for stream segregation - based upon distinguishing as stream of sound by its location and timbre - becomes challenged, as sounds are broken down in space and timbre and then re-united.

Though WFS was seen as the most suitable diffusion environment for realising this type of idea, the principle is independent of technology and the composer has also prepared an ambisonic version of the piece. However, as Victor Lazzarini notes, "some diffusion technologies will convey the ideas better and more precisely than others...". He also noted that the Doppler effect (which was gained automatically by the delay line based implementation) gave an extra element to the piece.

Immersive Motion Study (2007) - Shintaro Imai

Imai himself describes his use of spatialisation throughout his compositional work as:

> "I'm using spatialisation to make particular sound textures in space. In this case spatial reality is not so important, but I would need "high resolution" sound image as possible."

In *Immersive Motion Study*, Imai starts with drawing out the space from a single point, to surrounding the listener. During the composition, this space is kept defined through surrounding movement of several sources, building up to a climax, after which the listener is completely immersed in a vivid soundscape. Towards the end this soundscape then dissolves and retreats to the periphery. The technique he uses for this is moving "stereo-made sources in point-symmetry to make complex sound movement textures."

While working with the system, Imai found an interesting difference in sound character between plane waves and point sources, something which he would like to explore further a next time.

4.3.2 Works created on the IRCAM - Sonic Emotion system

sonicEmotion have organised several presentations in collaboration with the Musikforum in Stuttgart [Wit04, e.g], as well as realised several artistic works in collaboration with IRCAM and Centre Pompidou in Paris, such as for the DADA exhibition (see below), the Agora/Resonances festival[34] and the Samuel Beckett exhibition[35].

L'Amiral cherche une maison à louer (2005) - Gilles Grand

For the Dada exhibition in Centre Pompidou from October 5th, 2005, till January 9th, 2006, Gilles Grand[36] created a sound installation utilizing the Wave Field Synthesis system of *sonicEmotion*, in cooperation with IRCAM[37][Kuh06]. The sound installation was based on the poem (Simultangedicht) of Tzara, Huelsenbeck and Janco from 1916, and aimed to create a new interpretation of this poem, which was cited simultaneously by three actors in three languages, German, English and French. The installation was presented as a completely white room; the 56 speakers used were flat panel speakers, invisible for the audience. The three virtual actors move around through the space. In the realisation directional sources are used, so one can hear a difference depending on in which direction the virtual actor is turned.

4.3.3 Works created for the *The Game of Life* system

Wouter Snoei's *Correlation* (2006) is a composition constisting of two fairly independent parts. The piece is an exploration of different types of trajectories possible with WFS, such as sounds that move, spiralling into each other, random movements through the space, constellations of point sources around the audience in elliptic forms, which would change over time, melodies played from different locations (serial approach), as well as very fast movements along the array.

Yannis Kyriakides uses snippets of vocal sounds, that form words when they meet in his *Music in a foreign language* (2006).

Barbara Ellison works in *A net to catch contingency* (2006) with a drone that slowly moves through space, a static water sound in one corner and mainly circular movements of other sounds; vocal sounds play an important role in her piece as well, returning every now and again in the piece.

For the SuperCollider Symposium 2007, Tom Hall (*Skeletal Keys*), Alo Allik (*"Artificial Soul"*), Jeff Carey (*Structural Unit II*), Sergio Luque (*Happy Birthday*) and Robert van Heumen (*Phases2 (at the Symposium)*) created further compositions.

Tom Hall uses distinct points in space to spatialise piano and violin sounds

[34] "Seule avec Loup" (June 2006) - N+N Corsino Dance Group, which is an interactive choreographic world, with both 3D visuals and audio. `http://www.nncorsino.com/`.

[35] Centre Pompidou, Samuel Beckett exhibition, March 14 - June 25 2007

[36] `http://ouir.free.fr`

[37] `http://www.ircam.fr/99.html?event=314`

in a very quiet way. In his program notes he writes:

> "Skeletal Keys, a composition for wavefield synthesis playback, explores notions of harmonic and motivic cell repetition and reconfiguration found in the music of Erik Satie (1866 - 1925) and Morton Feldman (1926 - 1987). Such techniques were fundamental to both composers in different ways, from Feldman's early "graph" music and later notational practice, to Satie's early "mystical" works and the later pieces such as Cinema. Skeletal Keys contains elements of a number of these works, each inhabiting different spatial zones."

Alo Allik's piece, whose title translates to *Artificial Soul*, creates a subtle and quiet, almost meditative, spatial athmosphere, using WFS, where he makes use of his previous experiences with ambisonics.

Jeff Carey's piece is inspired by the chemical building up of molecules, how they are combined to larger units, and uses in his piece a simple FM synthesis unit as his "structural unit". Spatially he slowly builds up an encompassing sound field around the listener.

Sergio Luque's piece is based on extensions of Xenakis' stochastic theories. Spatially the piece is mostly frontal with some diversions to the left and right and some effects of depth.

Robert van Heumen establishes a dialog between sounds, and uses contrasts between close positions and surrounding ambience, from the program notes: "A blend of algoritmic composition & field recordings, inspired by the work of James Tenney".

4.3.4 SuperMono

Alex McMahon created the composition *Stasis*, "a spectral composition for string quartet and electronics was developed in conjunction with the system. This composition process explores the inherent structure of a simple sound to produce an absolute music. Synthesised sound sources are assigned three-dimensional spatial positions and trajectories. It is intended that the system should provide each member of the audience with a unique experience of the same live performance." [McM06].

The speaker array he used is a 4 by 4 grid of speakers, which were used to create "steerable lobes" to project sound onto the audience.

Chapter 5

Source model

5.1 Physics of sounding objects

5.1.1 Acoustic waves in solids

In a solid, sound can propagate both as a transverse wave, as well as a longitudinal wave. Propagation speed is dependent on the material characteristics of the solid: $v = \sqrt{\frac{B}{\rho}}$ where B is the bulk modulus and ρ is the density. The speed for transverse (waves of alternating shear stress) and longitudinal waves (waves of alternating pressure deviations from the equilibrium pressure, causing local regions of compression and rarefaction) can differ in different directions if the material is not isotropic. Dispersion occurs in most material media, so the wave energy is more spatially spread out and high and low frequency waves propagate at different speeds. In general, the speed of sound within a solid is much higher (5 to 20 times as high, depending on the material), than the velocity of sound in air. Upon reaching a surface or an edge of the solid, or a border between materials, some acoustical energy will be reflected, whereas other parts will transfer into the medium on the other side of the edge, border or surface. Thus, a complex pattern of surface vibrations on the object will occur, based on the material and the geometry of the object.

It can be observed that the vibration of the surface of the object is not a superposition of randomly decorrelated point sources, rather, the sound coming from different points on the surface has a (complex) physical relation to the vibration of the object at other locations.

5.1.2 Finite difference methods

Finite Difference Time Domain (FDTD) modelling is based on direct discretisation of the wave equation [KMP04]. The second-order partial derivatives are replaced by symmetric second-order differences, for both time and place. This results in a recursion formula for each spatial position where the two neighbouring nodes and the value of the node itself, one time step before, are used to compute the next value of the node. FDTD is based on computing Kirchhoff quantities. The advantage of the FDTD method is that only two unit delays per node are needed in any dimensionality, while the Digital Wave Guide (DWG) method needs $2 \times K$, where K is the dimensionality of the model. On the other hand, FDTD are numerically less robust.

5.1.3 Digital waveguide mesh method

The Digital Wave Guide (DWG) mesh method is a variant of the FDTD methods, but has its origins in the field of Digital Signal Processing (DSP) and is designed for efficient implementation. DWG methods are based on wave quantities. In a digital waveguide mesh (e.g. [BM04, Smi92]), points in space are viewed as units which are connected through bidirectional delay lines and scattering junctions which act as spatial and temporal sampling points within the modelled space. In higher dimensions different mesh topologies can be used, for example tetrahedral or rectilinear. For a lossless junction connecting lines of equal impedance, two conditions must be fulfilled; namely (1) the sum of inputs equals the sum of outputs (flow adds to zero), and (2) signals at each intersecting waveguide are equal at the junction (continuity of impedance) [Sav00].

Waveguide mesh models are limited by the dispersion error, i.e. the velocity of the propagating wave is dependent upon both its frequency and direction of travel, leading to wave propagation errors. This error is highly dependent on the chosen mesh topology, and can be compensated to some extent using frequency warping techniques. The accuracy of the method primarily depends on the density of the mesh.

5.1.4 Modal synthesis

In modal synthesis, a structure is treated as an assembly of substructures, each of which is analysed as a separate unit. The equations of motion of the complete structure are formulated by synthesising the properties of the components, such as mode shapes and interface compatibility conditions [Agr76].

Examples for application of this method for musical purposes, can be found in the software *Modalys* [IRC05, EIC95], in [Adr91], in [Bis00] and [HvdDF03]. This method can be used for realtime synthesis with current CPU power.

5.1.5 Combinations

The techniques described above can be used in combination with each other. *Modalys* provides a finite element method to calculate the vibration of a substructure [IRC03]. Bilbao [Bil06] describes a method to use digital waveguide extraction to calculate the modes for higher dimensional structures, while Karjalainen and Mäki-Patola [KMP04] describe combined methods for FDTD and DWG methods.

5.2 Source model for 3D objects

The wave field of a sounding object can be approximated by superposition of the wave fields of a number of point sources. The locations of these point sources and the number can be chosen arbitrarily, but it makes sense to choose locations on the surface of the object. This separates the calculation of the vibration of the object itself from the calculation of the radiation of the sound from the object into air.

In practice, the sound emitted from different points on the object's surface will be correlated. For a practical implementation it is useful to assume that for a point Ψ on the surface:

$$S_\Psi(\vec{r}_\Psi, \omega) = S(\omega)G(\vec{r}_\Psi, \omega) \tag{5.1}$$

i.e. from each point source that is part of the distribution, a source signal $S(\omega)$ filtered with G is emitted (G depending on the location $\vec{r}_\Psi$ of the point source Ψ and the angular frequency ω of the sound). Applied to the reproduction of electronic music, this allows a composer to determine the filter characteristics of his/her source object and the sound signal emitted by the source independently. Thus, the resulting filtering function for each speaker can be determined in advance and the signal input can be convolved in realtime using this filter.

5.2.1 Triangulated meshes

The geometry of the source object can be effectively modelled with a triangulated mesh. This is a common format used in computational and graphical modelling (e.g. [BB06]), that can equally be used for acoustic modelling. A triangulated mesh defines the geometry as a collection of triangular surfaces, edges and vertices (points). The vertices can be used as point sources in our model.

5.3 Discretisation

Spatial aliasing in WFS due to the discretisation of the secondary sources, i.e. the loudspeaker array, has been researched and documented in the literature. With an arbitrary shape of the primary source, spatial aliasing can also occur due to spatially sampling the wave field on the source surface. This sampling is not dependent on physical constraints, as in the case of the WFS reproduction with loudspeakers, where the speaker size provides the constraint. Instead the sampling distance, $\Delta\xi$, can be chosen freely. However, for a practical implementation it is desirable to keep the calculation power and data size requirements as low as possible, and thus, to choose $\Delta\xi$ as large as possible.

If we assume that for a correct reproduction, the speakers should be driven with an alias-free signal, we can derive the following formula for the maximum sampling distance ($\Delta\xi_{max}$) of a source (in analogy to the spatial aliasing fre-

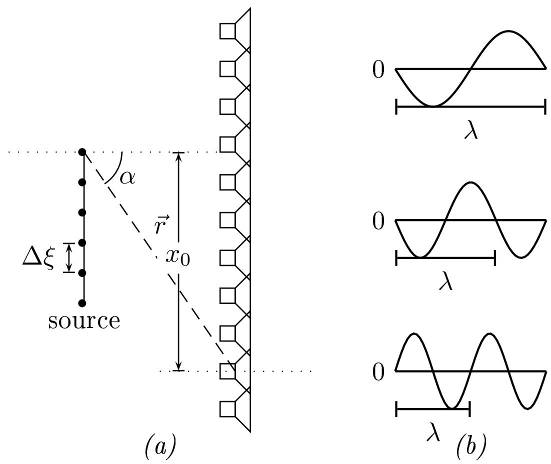


Figure 5.1: *a)* Geometry for the derivation of the aliasing formula (5.2). *b)* Examples of amplitude variation over the length of a line source with an indication of the corresponding λ, to determine the Rayleigh criterion (eq. (5.3)).

quency for the loudspeaker distance (eq. (2.37) and [Spo05])):

$$\Delta\xi_{max,1} = \frac{c}{f_{max}(1 + |\sin\alpha_{max}|)} \tag{5.2}$$

Where c is the velocity of sound, f_{max} is the highest frequency in the source signal, and α_{max} is the largest angle that the vector between source point and a speaker creates with the normal on the source surface (see figure 5.1a). For the geometry, as shown in this example, we can replace $\sin\alpha_{max}$ with $\frac{x_{0,max}}{r_{max}}$ and we can see that $\Delta\xi_{max}$ will be larger at greater distances from the array.

Another source for spatial aliasing can be the undersampling of the amplitude variations over the source surface. Here the Rayleigh criterion (e.g. [BCJ01]) determines the maximum sampling distance:

$$\Delta\xi_{max,2} = \frac{\lambda_{min}}{4} = \frac{1}{4k'_{max}} \tag{5.3}$$

here λ_{min} is the smallest wavelength of the amplitude variation (see figure 5.1b) and $k'_{max} = \frac{k_{max}}{2\pi}$ can be seen as its inverse, the spatial frequency of the amplitude.

The maximum sampling distance allowed is then the minimum of equations (5.2) and (5.3).

As with the arbitrarily shaped arrays discussed in section 2.6.4, this is not a fixed criterion: the object can have an irregular shape and thus the angle between the surface normal and the 'source point to speaker'-vector will be different for each face and each speaker.

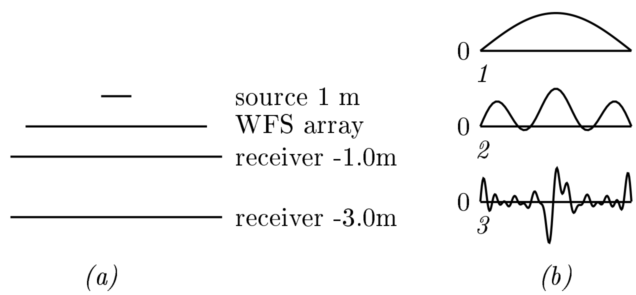


Figure 5.2: *a)* Geometry of the simulation experiments. *b)* Amplitude variations of the different sources, λ_{min}-values: 2.0, 0.40 and $0.069m$.

5.3.1 Simulations

To study discretisation effects, some simulations were done with *Matlab* [Mat]. A line source with a length of $1.0m$ was placed $1.0m$ behind the WFS speaker array, parallel to and centred to the centre of the WFS array. The source signal used was a Gaussian wavelet at two different frequency bandwidths (first with centre frequency $750Hz$ and maximum $3kHz$, second with $200Hz$ and $750Hz$ respectively; see figure 5.3.1). The source signal was extrapolated with a wave field extrapolation operator based on the Rayleigh I integral. In the simulation experiments 2 receiver distances were used: 1 and 3 meters in front of the array (see figure 5.2a). The reference line was at 3 meters in front of the array. The WFS reproduction array was sampled at $5cm$ intervals, as were the listening positions. The corresponding aliasing frequency for this array is $3.4kHz$; for both source wavelets this is above the highest frequency component. Thus, no spatial aliasing is introduced by the WFS reproduction array.

In figure 5.3 the effects of aliasing are shown. Already in the WFS reproduction signal, there are aliasing effects visible if source sampling discretisation $\Delta\xi$ is taken too large (right-hand plots: $\Delta\xi = 0.2m$), and aliasing appears as side lobes in the higher frequency parts (from ca. $1500Hz$). In the receiver area this aliasing should be audible mainly towards the sides of the listening area, while in the plots on the left-hand side ($\Delta\xi = 0.1m$), these effects are not visible.

In figure 5.4 the field extrapolated to the listening area is shown at $1m$ from the array, for a source $1m$ behind the array with an amplitude distribution of type 2 as indicated in figure 5.2b. For this source $\Delta\xi_{max,2} = 0.1m$. The reproduction is indeed correct for this value. At $0.2m$ the frequency spectrum becomes less focused and thus incorrect. In figure 5.5 we see similar plots for a source of type 3 ($\Delta\xi_{max,2} = 0.017m$); here the effects are much clearer: the reproduction with $\Delta\xi = 0.01m$ is correct, but at $0.05m$ the reproduction is no longer correct, even further away from the WFS array.

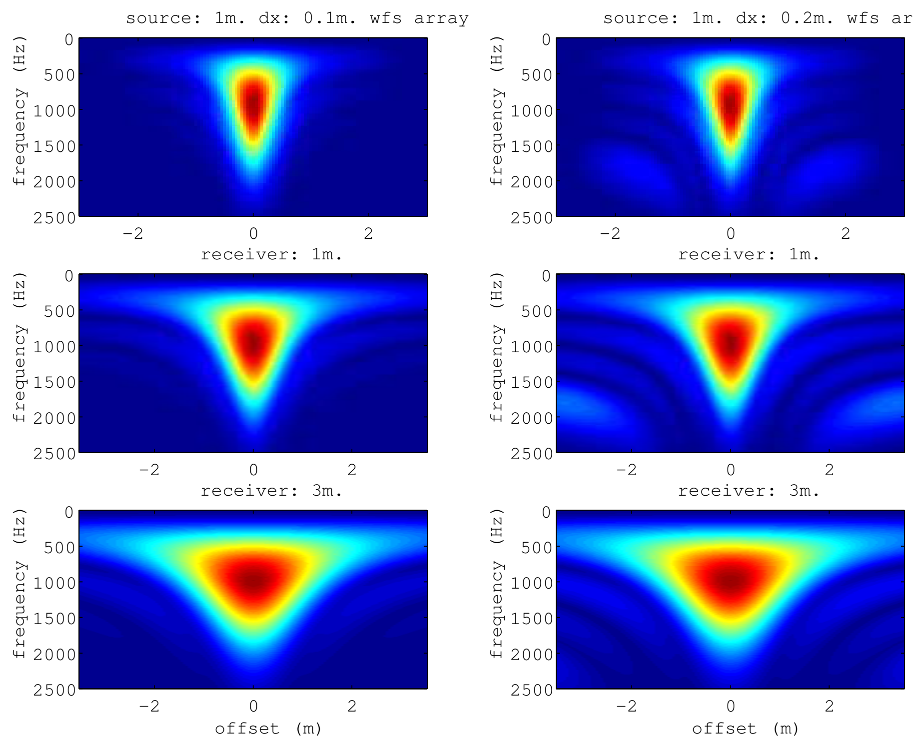


Figure 5.3: Aliasing effect for a source signal of type 1 (see figure 5.2b), with a Gaussian wavelet with centre frequency at 750 *Hz* and highest frequency at 3 *kHz*. The top plots are the WFS reproduction signals, the lower plots at receiver distance $1m$ and $3m$. Left-hand plots are with discretisation distance $0.1m$, the right-hand plots with $0.2m$.

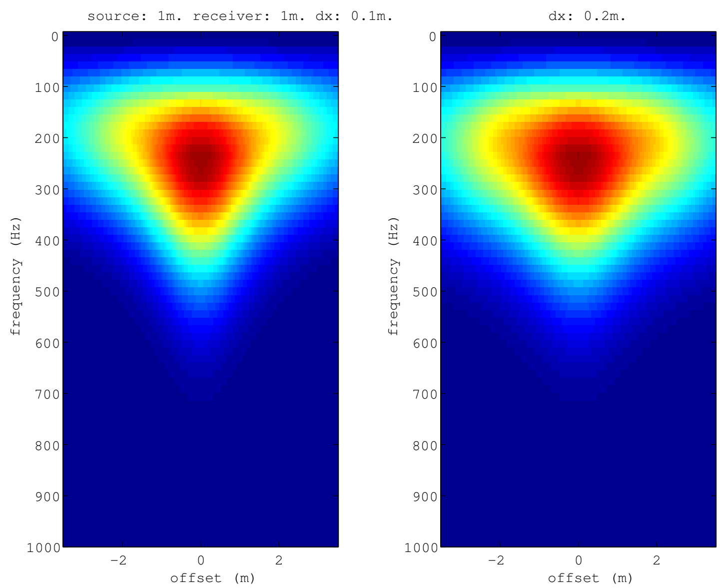


Figure 5.4: WFS reproduction at receiver line 1 meter in front of the array for source of type 2 (see figure 5.2b), low frequency wavelet, with $\Delta\xi = 0.1m$ (left) and $0.2m$ (right).

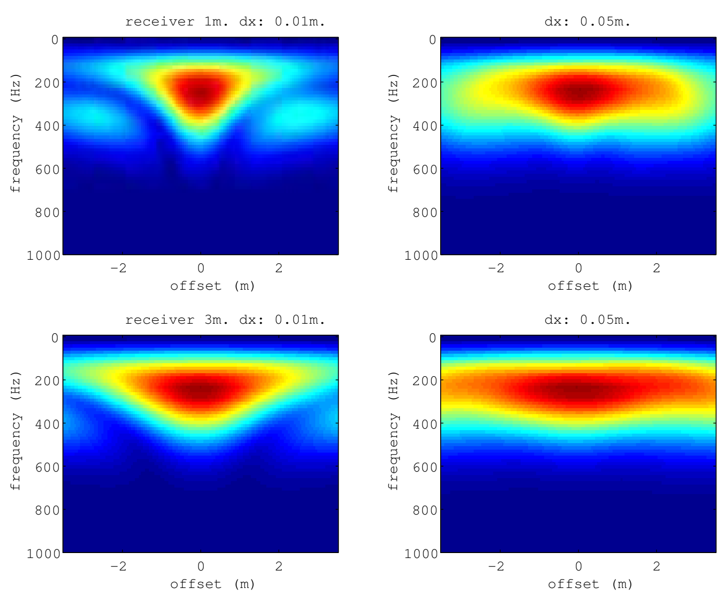


Figure 5.5: WFS reproduction at receiver line 1 (top) and 3 (bottom) meter in front of the array for a source of type 3 (see figure 5.2b), low frequency wavelet, with $\Delta\xi = 0.01m$ (left) and $0.05m$ (right).

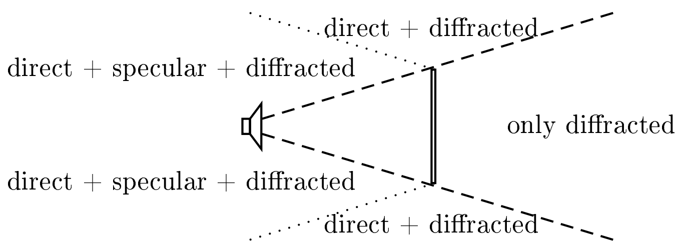


Figure 5.6: The problem of diffraction.

5.4 Diffraction

The sound emitted by the source object diffracts around the object itself.

The basic problem of diffraction is illustrated in figure 5.6. An object is irradiated by a source and blocks the transmission of the sound. We can distinguish different zones: the shadow zone, where no direct sound arrives, the specular zone, where both direct and reflected sound arrives and a zone where only direct sound arrives. In all zones diffracted sound arrives.

5.4.1 Overview of diffraction theories

In this section a short review will be given of various models to calculate diffraction of waves, motivating the choice for the model, that is used as the source model, and is described in detail in the remainder of the section.

5.4.1.1 Geometrical and Uniform Theory of Diffraction (GTD/UTD)

The Geometrical Theory of Diffraction (GTD), as stated by Keller [Kel62] in the 1960s is a description of diffraction in terms of rays. Starting from a generalised Fermat's principle, the GTD states the existence of diffracted rays by wedges and peaks, creeping rays, plus many others in addition to the classical direct and reflected rays of geometrical optics. The field at a receiver point in the shadow zone of an obstacle can then be predicted as the sum of all the fields of individual rays, provided that all paths from source to receiver are known.

There is a problem for the calculation of the field at the shadow boundary. This is solved with the Uniform Theory of Diffraction (UTD), that is derived from the GTD with the addition of higher order terms after Lüneberg-Kline [KP74].

5.4.1.2 Radiosity or Energetic models

Reboul et al. [RBPL05] describe a method based on radiosity. This method attempts to find a transfer equation which states the equilibrium of energy exchange between two facing surfaces. All rays travelling between these surfaces are taken into account in the form of an integral equation. In cases of multiple diffraction, this has the advantage that the problem can be solved in a finite number of steps, whereas methods based on the GTD have to be truncated at some point (i.e. the calculation only goes up to a fixed order of diffraction). Though the results of the method are promising, results in the vicinity of boundaries between illuminated and shadow zones are incorrect, just as in the GTD. In addition, interference effects cannot be described, as the energies are simply added. This is a limitation that does not occur in GTD.

Another method based on radiosity is described by Stephenson [Ste04], which is essentially a beam tracing method, where at suitable points in the calculation beams are recombined, so as to prevent an explosion of calculation time. Although he describes the method extensively, it seems he has not implemented it.

5.4.1.3 Biot-Tolstoy and Medwin's 'discrete Huygens interpretation'

Biot and Tolstoy [BT57] present explicit impulse response (IR) solutions for the problem of edge diffraction from an infinite wedge irradiated by a point source for the cases of a rigid wedge and a pressure release wedge.

Medwin et al. [MCJ82] present an interpretation of the Biot-Tolstoy model which they describe as a *discrete Huygens interpretation*, which can be used for finite wedges and can be extended to handle multiple diffraction. The model is a secondary source method, which assumes a number of secondary sources on the diffracting edge.

Svensson et al. [SF99] revised Medwin's theory and have since worked on solutions for singularities that occur in the integrand of the method [SC06]. Together with Calamia, several aspects of the numerical implementation have been presented [CS05, CSF05, CS07].

Of the three different models reviewed here, the last one can easily be integrated with the source model presented in this chapter, and results of the method have proven (in the literature) to be accurate. Therefore, this model will be discussed in more detail in this chapter and has also been implemented.

5.4.2 Secondary source model

The problem considered is that of a sound source irradiating a rigid or soft object [SF99]. The impulse response (IR) for plane-surfaced objects can be stated as a sum of the geometrical acoustics IR, h_{GA}, and the diffraction component, h_{diffr}. The direct sound and spectral reflections of first and higher orders will be contained in h_{GA}. In the case of an object with an entirely convex geometry (i.e. no indents), h_{diffr} will consist of only first and higher order diffraction

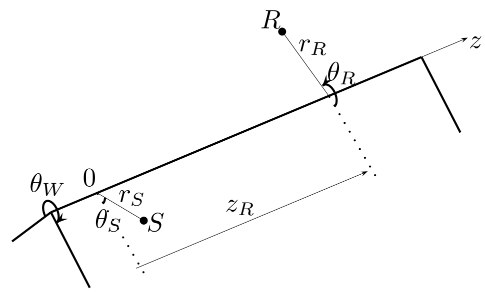


Figure 5.7: The geometry of a wedge irradiated by a point source S. Cylindrical coordinates are used with the z-axis along the edge of the wedge. The source has coordinates r_S and θ_S and is placed at $z_S = 0$. The receiver has the coordinates r_R, θ_R and z_R and the wedge has an open angle of θ_W.

components, whereas other geometries might cause combinations of specular reflections and edge diffraction. Here, we will only consider convex geometries; in section 5.6 will be described how the model can be extended to include specular reflections.

Consider a rigid wedge with a geometry as indicated in figures 5.7 and 5.8, where the cylindrical coordinates r_S, θ_S, 0 are used for the source and r_R, θ_R, z_R for the receiver. The continuous time edge diffraction IR can then be written as [SF99, CS05]:

$$h_{\text{diffr}}(\tau) = -\frac{\nu}{4\pi} \sum_{i=1}^{4} \int_{z_1}^{z_2} \delta(\tau - \frac{m+l}{c}) \frac{\beta_i}{ml} dz \tag{5.4}$$

where $v = \pi/\theta_w$ is the wedge index, θ_w is the open wedge angle, c is the speed of sound, and m and l are the distances from the source to the edge point and the receiver to the edge point, respectively. The integration limits z_1 and z_2 are the end points of the edge. The functions β_i are

$$\beta_i = \{\pm\}_i \frac{\sin(\nu\phi_i)}{\cosh(\nu\eta) - \cos(\nu\phi_i)} \tag{5.5}$$

where the function $\{\pm\}_i$ depends on the characteristics of the surface[1], and the angles ϕ_i are

[1] For hard material, the function is $+1$ for all i, for soft (pressure-release) material the values are: $i = 1 \rightarrow -1$, $i = 2 \rightarrow +1$, $i = 3 \rightarrow +1$ and $i = 4 \rightarrow -1$.

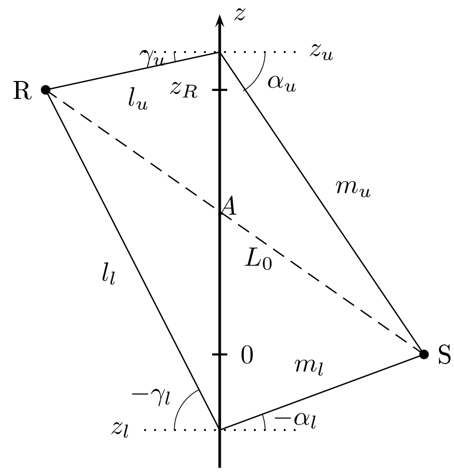


Figure 5.8: A plane view of the edge constructed from the two half-planes containing the edge and the source S and the edge and the receiver R, respectively. Two z-coordinates, z_l and z_u, are indicated for which the two sound paths $S - z_l - R$ and $S - z_u - R$ have identical path lengths. Also indicated is the shortest distance L_0, via the apex point, denoted A, of the edge. Angles are defined with signs so that $\sin \alpha = z/m$ and $\sin \gamma = (z - z_R)/l$.

$$\begin{aligned}
\phi_1 &= \pi + \theta_S + \theta_R \\
\phi_2 &= \pi + \theta_S - \theta_R \\
\phi_3 &= \pi - \theta_S + \theta_R \\
\phi_4 &= \pi - \theta_S - \theta_R
\end{aligned} \tag{5.6}$$

and the auxiliary function η is

$$\eta = \cosh^{-1}\{\frac{ml + (z - z_S)(z - z_R)}{r_S r_R}\} \tag{5.7}$$

5.4.2.1 Integrand singularities

The integrand in equation (5.4) has a singularity for certain receivers, namely near the zone boundaries, shown in figure 5.9. These singularities occur as the terms $\cosh(\nu\eta)$ and $\cos(\nu\phi_i)$, both take the value 1 for certain combinations of θ_S and θ_R. This behaviour occurs for z-values around the apex point, suggesting an approach with an analytical approximation of the integrand which is valid around the apex point. The derivation of this approximation is given in [SC06]; here only the result is given.

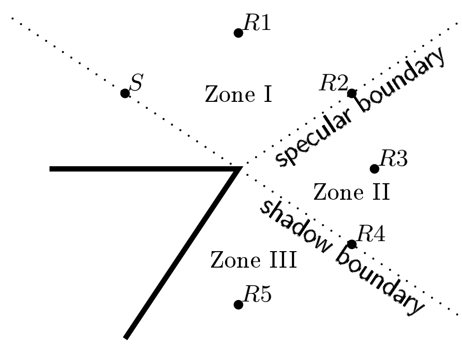


Figure 5.9: Zones defined by the geometrical acoustics as defined by the wedge and the source position S. A receiver ($R1$) in zone I will receive both reflected and direct sound, a receiver $R3$ in zone II only direct sound, and a receiver $R5$ in zone III neither. Diffracted sound will be audible in all three zones. Receivers $R2$ and $R4$ are on the specular and shadow boundary, respectively, for which $\theta_r = \pi \mp \theta_S$. For these cases the singularities in the integrand of equation (5.4) occur and the integrand should be approximated by equation (5.13).

Given an edge, source and receiver, the z-coordinate of the apex point is

$$z_a = \frac{z_R r_S + z_S r_R}{r_S + r_R} \tag{5.8}$$

and an approximation of the integrand β_i/ml near the apex point can be made by using a z-coordinate relative to z_a, $z_{rel} = z - z_a$.

For small values of z_{rel}

$$\cosh(\nu\eta) \approx 1 + \nu^2 \frac{(1+\rho)^4}{2\rho^2 R_0^2} z_{rel}^2 \tag{5.9}$$

and

$$ml \approx \frac{R_0^2 \rho}{(1+\rho)^2} + \frac{\cos\psi R_0(\rho - 1)}{\rho + 1} z_{rel} + \frac{\sin^2\psi(1+\rho)^2 - 2\rho}{2\rho} z_{rel}^2 \tag{5.10}$$

where the dimensionless quantity ρ is defined as

$$\rho = \frac{r_R}{r_S} \tag{5.11}$$

and the angle ψ, shown in figure 5.10, is defined such that

$$\sin\psi = \frac{r_S + r_R}{R_0}, \cos\psi = \frac{z_R - z_S}{R_0} \tag{5.12}$$

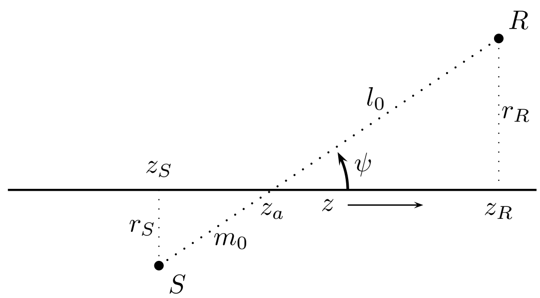


Figure 5.10: An unfolded 2D view of a wedge, source and receiver, showing the angle ψ. $R_0 = m_0 + l_0$ for equation (5.12).

Using these equations, the integrand β_i/ml can be approximated as

$$\frac{\beta_i}{ml} = B_0 \cdot \frac{1}{z_{rel}^2 + B_1} \cdot \frac{1}{z_{rel}^2 + B_2 z_{rel} + B_3} \tag{5.13}$$

where

$$\begin{aligned}
B_0 &= \frac{4R_0^2\rho^3 \sin(\nu\phi_i)}{\nu^2(1+\rho)^4[(1+\rho)^2\sin^2\psi - 2\rho]} \\
B_1 &= \frac{4R_0^2\rho^2\sin^2(\frac{\nu\phi_i}{2})}{\nu^2(1+\rho)^4} \\
B_2 &= -\frac{2R_0(1-\rho)\rho\cos\psi}{(1+\rho)[(1+\rho)^2\sin^2\psi - 2\rho]} \\
B_3 &= \frac{2R_0^2\rho^2}{(1+\rho)^2[(1+\rho)^2\sin^2\psi - 2\rho]}
\end{aligned} \tag{5.14}$$

Note that only B_0 and B_1 are functions of ϕ_i. The finite integral in equation (5.4) can be solved analytically when the integrand is given by equation (5.13). The solution will be discussed when we treat the numerical implementation in subsection 5.4.3.

5.4.2.2 Higher order diffraction

For a second order diffraction, where the sound diffracted by one edge is diffracted by another edge, we can write in analogy to equation (5.4) [SF99]:

$$h_{\text{diffr}}(\tau) = \frac{\nu_1\nu_2}{(4\pi)^2} \int_{z_{1,1}}^{z_{1,2}} \int_{z_{2,1}}^{z_{2,2}} \delta(\tau - \frac{m_1(z_1) + m_2(z_1,z_2) + l(z_2)}{c}) \frac{\sum_{i=1}^4 \beta_{i,1} \sum_{i=1}^4 \beta_{i,2}}{2m_1(z_1)m_2(z_1,z_2)l(z_2)} dz_1 dz_2 \tag{5.15}$$

Where z_1 is the integration variable on the first edge, and z_2 on the second edge. The factor 2 in the denominator compensates for the doubling in pressure generated by an acoustic source when it is mounted on a baffle. It should be 1 if the path from the first edge to the second edge does not run along a plane. It should be noted that $\beta_1[\alpha_1(z_1), \gamma_1(z_1, z_2), \theta_{S1}, 0]$ and $\beta_2[\alpha_2(z_1, z_2), \gamma_1(z_2), 0, \theta_{R2}]$, i.e. they are both dependent on both z_1 and z_2. In this formulation it has been assumed that the path is along a plane, so that $\theta_{R1} = 0$ and $\theta_{S2} = 0$.

Equation (5.15) can also be written as:

$$\begin{aligned} h_{\text{diffr}}(\tau) &= -\frac{\nu_1}{4\pi} \int_{z_{1,1}}^{z_{1,2}} [\frac{\delta(\tau - \frac{m_1(z_1)}{c})}{m_1(z_1)} * I_2] dz_1 \\ I_2 &= -\frac{\nu_2}{4\pi} \frac{1}{2} \int_{z_{2,1}}^{z_{2,2}} \delta(\tau - \frac{m_2(z_1, z_2) + l(z_2)}{c}) \frac{\sum_{i=1}^{4} \beta_{i,1} \sum_{i=1}^{4} \beta_{i,2}}{m_2(z_1, z_2) l(z_2)} dz_2 \end{aligned} \tag{5.16}$$

where $*$ denotes convolution. For higher order diffraction we can then generalise the result for arbitrary order N to

$$\begin{aligned} h_{\text{diffr}}(\tau) &= -\frac{\nu_1}{4\pi} \int_{z_{1,1}}^{z_{1,2}} [\delta(\tau - \frac{m_1(z_1)}{c}) \frac{1}{m_1(z_1)} * I_2(\tau)] dz_1 \\ I_2 &= -\frac{\nu_2}{4\pi} \frac{1}{2} \int_{z_{2,1}}^{z_{2,2}} [\delta(\tau - \frac{m_2(z_1, z_2)}{c}) \frac{\sum_{i=1}^{4} \beta_{i,1}}{m_2(z_1, z_2)} * I_3(\tau)] dz_2 \\ &\dots \\ I_N &= -\frac{\nu_N}{4\pi} \frac{1}{2} \int_{z_{N,1}}^{z_{N,2}} \delta(\tau - \frac{m_N(z_{N-1}, z_N) + l(z_N)}{c}) \frac{\sum_{i=1}^{4} \beta_{i,N-1} \sum_{i=1}^{4} \beta_{i,N}}{m_N(z_{N-1}, z_N) l(z_N)} dz_N \end{aligned} \tag{5.17}$$

Alternately, we can write this from a receiver point of view:

$$\begin{aligned} h_{\text{diffr}}(\tau) &= -\frac{\nu_N}{4\pi} \frac{1}{2} \int_{z_{N,1}}^{z_{N,2}} [\delta(\tau - \frac{l_N(z_N)}{c}) \frac{1}{l_N(z_N)} * I_{N-1}(\tau)] dz_N \\ I_{N-1}(\tau) &= -\frac{\nu_{N-1}}{4\pi} \frac{1}{2} \int_{z_{N-1,1}}^{z_{N-1,2}} [\delta(\tau - \frac{m_N(z_{N-1}, z_N)}{c}) \frac{\sum_{i=1}^{4} \beta_{i,N}}{m_N(z_{N-1}, z_N)} * I_{N-2}(\tau)] dz_{N-1} \\ &\dots \\ I_1(\tau) &= -\frac{\nu_1}{4\pi} \int_{z_{1,1}}^{z_{1,2}} \delta(\tau - \frac{m_1(z_1) + m_2(z_1, z_2)}{c}) \frac{\sum_{i=1}^{4} \beta_{i,1} \sum_{i=1}^{4} \beta_{i,2}}{m_1(z_1) m_2(z_1, z_2)} dz_1 \end{aligned} \tag{5.18}$$

This last formulation is useful for our source model, as it enables us to calculate the integral I_{N-1} for the source object as a whole, as the source points on the object are static with respect to the diffracting edges also on the object. Then, only the last integral needs to be calculated when we know the source object position relative to the loudspeaker array. This is further clarified in section 6.6.

5.4.3 Numerical implementation

The transition from the continuous time domain to discrete time domain can be achieved by subdividing the edge into segments, and calculate for each segment the contribution Δh_i to the IR $h_{\mathrm{IR}}(n)$[2] and distributing this contribution among the appropriate time samples. In general, the numerical integration is straightforward, but special care needs to be taken for the singularity as described in subsection 5.4.2.1. Regardless of whether or not the singularity occurs, the analytic approximation of equation (5.13) is valid for the first sample of the IR, $h(n_0)$. Thus, the approximation can always be used for the contribution around the apex point.

5.4.3.1 Edge subdivision strategies

Sample-aligned segments

In the case of sample-aligned segments the edge is divided into segments, each corresponding to the contribution to one sample of the IR [CS05]; this means that the integration limits $z_{n,1}$ and $z_{n,2}$ in

$$h_{\mathrm{diffr}}(n) = -\frac{\nu}{4\pi}\sum_{i=1}^{4}\int_{z_{n,1}}^{z_{n,2}}\frac{\beta_i}{ml}dz \tag{5.19}$$

are determined by the source and receiver locations such that the path length $m_n + l_n = c(n \pm 0.5)/F_S$, where c is the sound velocity, F_S is the sampling frequency and n is the sample index. The edge segment lies between intersections of the edge and two confocal ellipsoids, of which the foci are the source and receiver locations and whose axes lengths are determined by the path lengths, as shown in figure 5.11.

Using sample aligned segments has several advantages:

1. By choosing a high enough sampling frequency, the spectrum of the discrete IR can match up to the continuous IR up to a chosen frequency, dependent on the inherent low pass filtering effect of area sampling of equation (5.19).
2. For the onset sample n_0 the analytical approximation of equation (5.13) can easily be used. It should be noted that even when the approximation is not necessary, it is beneficial to isolate the processing of the onset sample, as the diffracted energy at the start of the IR is very high, and thus inaccuracies will have greater effect on the accuracy of the whole calculation.
3. Two sample-aligned segments that contribute to the same sample (so on each side of the apex point) do so equally. Thus it is sufficient to define and process only the samples of the longest branch and double the contribution for these segments if a corresponding segment on the other branch exists.

[2]throughout the following I will keep to the convention of using $h_{\mathrm{IR}}(n)$ for the full IR, and $h(n)$ for the contribution at the sample n of the IR.

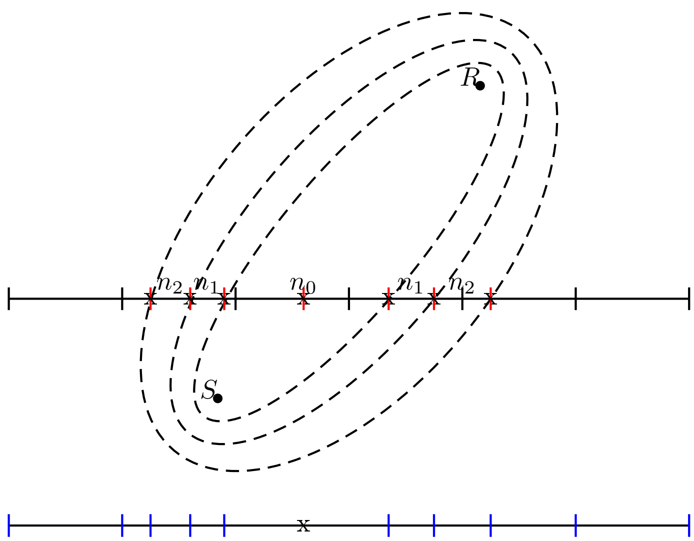


Figure 5.11: Unfolded 2D view of a source, receiver, and segmented edge. The upper edge is marked with the boundaries for a 3-sample alignment zone (samples n_0, n_1, and n_2) in black and the original even-segment boundaries in red. The lower edge (S and R not shown) is marked with the modified segment boundaries for the hybrid subdivision scheme in blue: even segments overlapping the alignment zone have been truncated at the edges of the zone, and those completely within the alignment zone have been discarded.

4. The method is straightforward, and can use both numerical integration methods, as well as analytical approximation.

However, this method also has disadvantages:

1. The segment boundary calculations are time consuming, and a high sampling frequency can lead to a very large number of segments.
2. The boundaries must be recalculated for changes in either source or receiver position.
3. The boundaries are only sample aligned for first-order diffraction, so the benefits do not extend to higher orders of diffraction.

Evenly sized segments

Alternatively, the edge (with length L) can be divided into evenly sized segments by setting a maximum segment size of Δz_{max} and subdividing the edge into k segments of length Δz, where $k = \lceil L/\Delta z_{max} \rceil$[3] and $\Delta z = L/k$ [CS05]. Associated boundaries are easy to calculate and are independent of the source and receiver locations, while the segments can be used for higher order calculations.

An edge segment j placed at position z_j will give a contribution Δh_j to the IR

$$\Delta h_j \approx -\frac{\nu}{4\pi} \sum_{i=1}^{4} \frac{\beta_i}{m(z_j)l(z_j)} \Delta z \tag{5.20}$$

This contribution should be added to a single time sample $n = f_s \frac{(m_j + l_j)}{c}$ or divided over two or more consecutive time samples:

$$\Delta h_j(n) \approx -\frac{\nu}{4\pi} \sum_{i=1}^{4} \frac{\beta_i}{m(z_j)l(z_j)} \Delta z W(n) \tag{5.21}$$

where $W(n)$ is a weighting function to divide the value over the relevant samples, and only has values between the time samples corresponding to the paths through $z_j \pm \Delta z$ for a non-apex segment. For apex segments, the boundaries are the onset sample and the sample corresponding to the longest path through either $z_j \pm \Delta z$.

If the path lengths $z_j \pm \Delta z$ are denoted as p_a and p_b with $p_b > p_a$, corresponding to the real values sample numbers $N_a = p_a \cdot F_s/c$ and $N_b = p_b \cdot F_s/c$, and integer sample numbers $n_a = \mathrm{round}(N_a)$ and $n_b = \mathrm{round}(N_b)$, then the sample span for the segment is $S_{sp} = N_b - N_a$, which comprises $S_I = n_b - n_a - 1$ complete inner samples and 2 fractional outer samples. For the latter, the segment fractions "covered" are $\alpha_a = n_a + 0.5 - N_a$ and $\alpha_b = N_b - n_b + 0.5$. A simple flat distribution of Δh_j over the sample span results in $\alpha_a \Delta h_j / S_{sp}$ in sample n_a, $\alpha_b \Delta h_j / S_{sp}$ in sample n_b and $\Delta h_j / S_{sp}$ in each of the S_I inner samples. This is illustrated in figure 5.12.

[3] $\lceil f \rceil$ means the value of f rounded up to the next integer

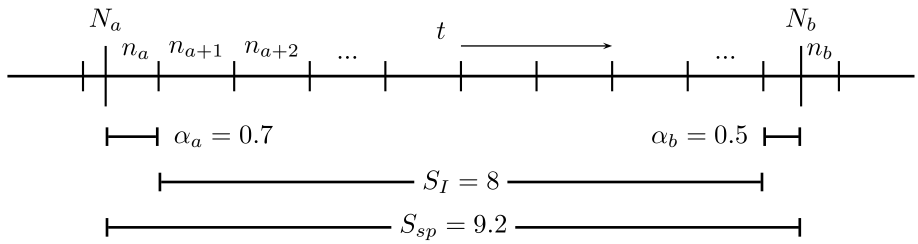


Figure 5.12: Illustration of the weighted distribution of the contribution over the available samples.

However, a simple flat distribution leads to a staircase effect in the impulse response. To remedy this, it is possible to choose a different weighting function, which adjusts the contributions based on the local slope of the IR. For example, if two adjacent multi-sample segments with total amplitudes Δh_1 and Δh_2, and sample spans $S_{sp,1}$ and $S_{sp,2}$. With the flat distribution, the middle sample of the first segment would be given value $\Delta h_1/S_{sp,1}$ and the middle sample of the second segment value $\Delta h_2/S_{sp,2}$. Under the assumption that the IR is locally linear, the slope s between these samples can be calculated and the distribution of Δh_2 can be adjusted such that the slope over $S_{sp,2}$ is equal to s.

The disadvantages to evenly sized segments are, that a small value of Δz may lead to a large number of segments to process, and the per-segment processing is somewhat more complicated than with sample-aligned subdivision, as for each segment the sample span must be calculated and the contribution must be distributed. A large value of Δz may lead to inaccuracy, though the slope correction has proven to be a rather good approximation. Finally, the location of the onset sample is not given automatically, which makes it necessary to include an extra check, to confirm whether the apex is included in the segment, in order to avoid the onset singularity and to calculate the correct onset sample.

Hybrid subdivision strategy

A hybrid strategy can be used to benefit from the advantages of both the sample aligned segments and evenly sized segments. In this strategy [CS07], sample aligned segments are used for the first M samples (including the apex segment, which uses the analytical approximation for the integrand) and evenly sized segments are used for the rest of the edge. Any portion of an evenly sized segment that overlaps with a sample aligned segment is discarded.

5.4.3.2 Integrand singularities

Here we continue the discussion started in subsection 5.4.2.1, to discuss the analytical approximation of the integral in equation (5.19).

The approximation is needed only for the first sample of the edge-diffraction IR, $h(n_0)$, as this is the only sample affected by the singularity in the original integrand [SC06]. Thus, the limits of the integration will be the z-values that delineate the portion of the edge that contributes to the onset sample. Using

z-coordinates relative to the apex point, one of these values will be negative (z_0^-) and the other positive (z_0^+), so we can write for the first sample:

$$\begin{aligned} h_{\text{diffr}}(n_0) &= -\frac{\nu}{4\pi}\sum_{i=1}^{4}\int_{z_0^-}^{z_0^+}\frac{\beta_i}{ml}dz \\ &= -\frac{\nu}{4\pi}(\sum_{i=1}^{4}\int_{z_0^-}^{0}\frac{\beta_i}{ml}dz + \sum_{i=1}^{4}\int_{0}^{z_0^+}\frac{\beta_i}{ml}dz) \end{aligned} \tag{5.22}$$

and as two segments on opposite sides of the apex point, which contribute to the same sample contribute an equal amount,

$$h_{\text{diffr}}(n_0) = -\frac{\nu}{2\pi}\sum_{i=1}^{4}\int_{0}^{z_0^+}\frac{\beta_i}{ml}dz \tag{5.23}$$

To approximate this integral, we need to distinguish between a symmetrical and an asymmetrical case.

Symmetrical case

For a symmetrical case, either $z_S = z_R$, i.e. $\psi = \pi/2$, or $r_S = r_R$, i.e. $\rho = 1$, and in both cases $B_2 = 0$ in equation (5.13), thus the integral for the first sample becomes:

$$h_i(n_0) \approx -\frac{\nu}{2\pi}\cdot\frac{B_0}{B_3 - B_1}\int_0^{z_{range}}\frac{1}{z_{rel}^2 + B_1} - \frac{1}{z_{rel}^2 + B_3}dz_{rel} \tag{5.24}$$

The result of this integration becomes:

$$h_i(n_0) \approx -\frac{\nu}{2\pi}\cdot\frac{B_0}{B_3 - B_1}[\frac{1}{\sqrt{B_1}}\arctan\frac{z_{range}}{\sqrt{B_1}} - \frac{1}{\sqrt{B_3}}\arctan\frac{z_{range}}{\sqrt{B_3}}] \tag{5.25}$$

A special case for the symmetrical situation occurs when $\rho = 1$ and $\psi = \pi/4$. The integrand β_i/ml then simplifies to

$$\frac{\beta_i}{ml} \approx B_4\frac{1}{z_{rel}^2 + B_1} \tag{5.26}$$

where

$$B_4 = \frac{B_0}{B_3} = \frac{\sin(\nu\phi_i)}{2\nu^2} \tag{5.27}$$

The result of integration is then

$$h_i(n_0) \approx -\frac{\nu}{2\pi}\cdot\frac{B_4}{\sqrt{B_1}}\arctan\frac{z_{range}}{\sqrt{B_1}} \tag{5.28}$$

Asymmetrical case

The integral form for the ith term of the first sample of the diffraction IR is after some rearrangement

$$h_i(n_0) \approx \frac{\nu}{2\pi}\frac{B_0B_2}{B_1B_2^2+(B_1-B_3)^2} \int_0^{z_{range}}[\frac{z_{rel}+(B_1-B_3)/B_2}{z_{rel}^2+B_1} - \frac{z_{rel}+(B_1-B_3+B_2^2)/B_2}{z_{rel}^2+B_2z_{rel}+B_3}]dz_{rel} \tag{5.29}$$

The result of the integration is

$$\begin{aligned} h_i(n_0) \approx & \frac{\nu}{2\pi}\frac{B_0B_2}{B_1B_2^2+(B_1-B_3)^2}[\frac{1}{2}\ln|\frac{B_3(z_{range}^2+B_1)}{B_1(z_{range}^2+B_2z_{range}+B_3)}| \\ + & \frac{B_1-B_3}{\sqrt{B_1}B_2}\arctan(\frac{z_{range}}{\sqrt{B_1}})+\frac{2(B_3-B_1)-B_2^2}{2B_2}F] \end{aligned} \tag{5.30}$$

where F can take one of four forms, depending on the quantity

$$q = 4B_3 - B_2^2 \tag{5.31}$$

For $q < 0$ and finite, form I should be used, where

$$F_I = \frac{1}{\sqrt{-q}}\ln(|\frac{2z_{range}+B_2-\sqrt{-q}}{2z_{range}+B_2+\sqrt{-q}}\cdot\frac{B_2+\sqrt{-q}}{B_2-\sqrt{-q}}|) \tag{5.32}$$

For $q > 0$ and finite, form II should be used, where

$$F_{II} = \frac{2}{\sqrt{q}}(\arctan\frac{2z_{range}+B_2}{\sqrt{q}} - \arctan\frac{B_2}{\sqrt{q}}) \tag{5.33}$$

When $q = 0$, the third form is

$$F_{III} = \frac{4z_{range}}{B_2(2z_{range}+B_2)} \tag{5.34}$$

The fourth form $F_{IV} = 0$ is used when the denominator of q goes to zero. When q is written as

$$q = \frac{4R_0^2\rho^2[2(1+\rho^2)-\cos^2\psi(1+6\rho+\rho^2)]}{(1+\rho)^2[(1+\rho)^2\sin^2\psi-2\rho]} \tag{5.35}$$

it is clear that the denominator will be zero when

$$\sin^2\psi = \frac{2\rho}{(1+\rho)^2} \tag{5.36}$$

or, equivalently,

$$\rho = \cot^2\psi\cdot(1\pm\sqrt{1-\tan^4\psi}) \tag{5.37}$$

Choice of z_{range}

For the implementation, an appropriate value of z_{range} must be chosen. There can be two cases:

1. The whole segment from z_0^- to z_0^+ is contained in the edge, which is the most common case at high sampling frequencies.
2. One or both of the onset segment boundaries is not included in the edge. In this case the doubling of the integrand (see eq. (5.23)) is not valid for part of the segment. If $z_{edge}^{\pm}$ is the edge end point in either the positive or negative direction, then the smallest of $z_{edge}^{\pm} < z_0^{\pm}$ is taken as the initial choice for z_{range}, and uses, for the remainder of the calculation, a numerical integration.

Furthermore, the choice of z_{range} must satisfy the condition that z_{range} is small compared to m_0 and l_0 for the analytical approximation to be valid, so a criterion is for example $z_{range} = 0.05 \times \min(m_0, l_0)$. If the chosen value of z_{range} is smaller than either of the segment boundaries $z_0^{\pm}$, numerical integration must be used for the remainder of the segment.

5.4.3.3 Numerical integration

For the numerical integration of the integrand β_i/ml several techniques can be used. The most simple is a 1-point midpoint integration, which simply evaluates the integrand at midpoint and uses this multiplied with the segment size Δz as the resulting value. Alternate methods are a standard 3-point Simpson's rule integration, or a compound Simpson's rule integration with one step of Richardson extrapolation (also called a Romberg integration).

For three evenly spaced points x_0, x_1 and x_2 with distance Δx, the 3-point Simpson's rule states that [Wei05a]:

$$\int_{x_0}^{x_2} f(x)dx = \int_{x_0}^{x_0+2\Delta x} f(x)dx \approx \frac{\Delta x}{3}(f(x_0) + 4f(x_1) + f(x_2)) \tag{5.38}$$

A compound version of this rule can be obtained if we subdivide the interval $[x_0, x_1]$ again in n steps, provided we divide them in an even number of subintervals ($n = 2m$):

$$\begin{aligned}
\int_{x_0}^{x_1} f(x)dx \approx \\
& \frac{\Delta x}{3}\sum_{i=0}^{m-1}(f(x_0+2i\Delta x) + 4f(x_0+(2i+1)\Delta x) + f(x_0+(2i+2)\Delta x)) \\
= & \frac{\Delta x}{3}[f(x_0) + 4f(x_0+\Delta x) + 2f(x_0+2\Delta x) + ... \\
& ... + 2f(x_0+(n-2)\Delta x) + 4f(x_0+(n-1)\Delta x) + f(x_1)]
\end{aligned} \tag{5.39}$$

The accuracy can be written with Richardson extrapolation [Fen06] as

$$f_{exact}(x) = f(x;\Delta x) + \alpha\Delta x + ... \tag{5.40}$$

where $f(x;\Delta x)$ means the solution obtained at x with step size Δx, and where higher order terms are neglected. We do not know the coefficient α or the exact solution $f_{exact}(x)$. However, if we carry out a solution for two different time steps, we can solve the pair of equations and obtain an approximation for $f_{exact}(x)$ by truncating the series. The coefficient α is of no interest.

$$f_{exact}(x) \approx \frac{f(x;\Delta x_2)\Delta x_1 - f(x;\Delta x_1)\Delta x_2}{\Delta x_1 - \Delta x_2} \tag{5.41}$$

When $\Delta x_2 = \Delta x_1/2$ we obtain

$$f_{exact}(x) \approx 2f(x;\Delta x/2) + f(x;\Delta x) \tag{5.42}$$

In the case of integration by the compound Simpson rule, the order of the neglected terms is $O(\Delta x^4)$. Applying the method here gives the scheme (this form is known as the Romberg integration):

$$f_{exact}(x) \approx f(x;\Delta x/2) - \frac{f(x;\Delta x/2) - f(x;\Delta x)}{15} \tag{5.43}$$

5.4.3.4 Higher order diffraction

The integral formulations in (5.18) and (5.19) can be written in a discrete form; using the receiver formulation, we have

$$\begin{aligned}
h(n) &\approx -\frac{\nu_N}{4\pi}\frac{1}{2}\sum_{j=1}^{M_N}[\frac{W_N(n)}{l(z_{j,N})} * I_{N-1}(n)]\Delta z_{j,N} \qquad (5.44)\\
I_{N-1}(n) &= -\frac{\nu_{N-1}}{4\pi}\frac{1}{2}\sum_{j=1}^{M_{N-1}}[\frac{W_{N-1}(n)\sum_{i=1}^{4}\beta_{i,N}}{m_N(z_{j,N-1},z_{j,N})} * I_{N-2}(n)]\Delta z_{j,N-1}\\
&\dots\\
I_1(n) &= -\frac{\nu_1}{4\pi}\sum_{j=1}^{M_1}\frac{W_1(n)\sum_{i=1}^{4}\beta_{i,1}\sum_{i=1}^{4}\beta_{i,2}}{m_1(z_{j,1})m_2(z_{j,1},z_{j,2})}\Delta z_{j,1}
\end{aligned}$$

Here the functions $W_k(n)$ are the weight functions to divide the contributions over the relevant samples for order k, in analogy to the function $W(n)$ in equation (5.21).

Higher order diffraction is most easily implemented for evenly sized segments, though (with some extra effort) a hybrid approach can also be used. In this way, onset samples for each intermediary impulse response will be calculated correctly and inaccuracies will not propagate throughout the calculation.

5.5 The source model including diffraction

The source model described in section 5.2 can now be extended with calculations for diffraction around the edges of the three dimensional object. So for each point on the surface of the object, we calculate the WFS driver functions for each speakers of the direct sound, and then calculate the diffracted sound arriving at each speaker, using the method described in the previous section. Two things need to be taken into account:

- If we combine this formulation with the WFS operator for the secondary sources, we must replace the amplitude factor caused by the path l, with the WFS attenuation. The delay as calculated in the diffraction algorithm is the same as the WFS delay.
- As the sound from the diffracted source points for a convex object travel along the object's surface, a factor $1/2$ has to be added for the first order diffraction, as the source is as if it were placed in an infinite baffle (compare the discussion in [SF99] about the secondary sources which are used as sources for higher order diffraction).

The details of the implementation will be described in the next chapter.

5.6 Integrated modeling approach

The determination of the diffraction parameters can be used to detect whether a source point is occluded for a receiver, and whether specular reflections occur [CSF05]. The approach is as follows: from the triangulated mesh of the sound object we calculate and store the diffraction parameters in a first pass of the list of edges.

In a second pass over the list of faces, we evaluate the diffraction parameters to determine whether or not a face obstructs the direct sound, or creates a specular reflection. This is done by maintaining two counters for each face:

1. the number of edges for which $\phi_2 < 0$ or $\phi_3 < 0$,
2. the number of edges for which $\phi_4 > 0$.

When the first counter is 3 (i.e. for all of the edges the situation occurs), the face obstructs the direct sound. Once a face has been found to obstruct the sound for a sound source, no further faces need to be checked for direct sound obstruction.

When the second counter is 3, the face creates a specular reflection. Further evaluation of parameters is necessary to find the reflection point. Once a specular reflection has been found to be created by a face, no other faces in the same plane need to be tested.

Both cases will be discussed in more detail below.

5.6.1 Direct sound

Relative to a single edge, a face will occlude the direct-sound path if the angles θ_S and θ_R differ more than π, i.e. if $\pi - |\theta_S - \theta_R| < 0$. If this is the case for all

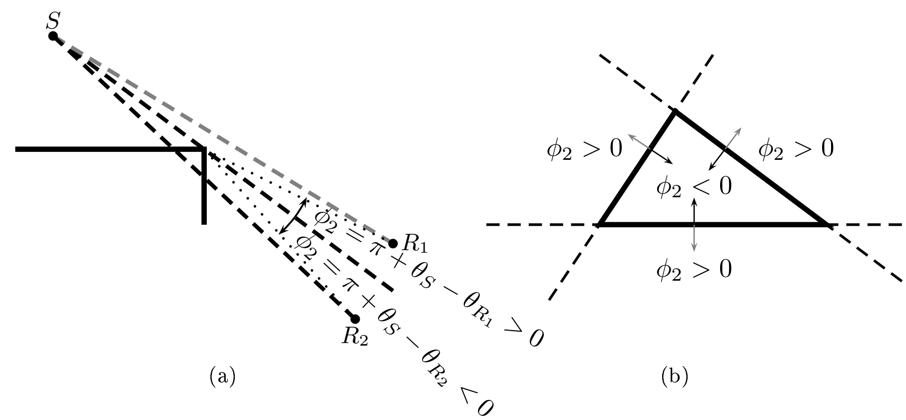


Figure 5.13: Checking for the occlusion of direct sound. (a) the face tested is shown as a horizontal line, and the edge comes out of the page. The diffraction parameter ϕ_2 from equation (5.6) measures the angular distance of the receiver from the shadow boundary. If $\phi_2 < 0$, for example for R_2, the sound source is occluded by the edge for the receiver. (b) If $\phi_2 < 0$ for all three edges of the face, the face occludes the direct sound.

three edges of a triangular face, that face occludes the sound. This is illustrated in figure 5.13.

The result can be generalised to a face that is bordered by an arbitrary polygon: if for all edges $\pi - |\theta_S - \theta_R| < 0$, then the face occludes the direct sound. Inversely, any edge for which $\pi - |\theta_S - \theta_R| > 0$, disqualifies the face as a possible occluder (so no more edges need to be checked).

This calculation is a fairly cheap method to check the visibility of a source point. Another method would be to calculate the intersection point of the source-receiver path with each surface element of the object; if this point is within a face, then the point is not visible. More detail on this method will be given in the next subsection, where it will be used to find the reflection point.

5.6.2 Specular reflections

The test for a specular reflection involves testing whether $\phi_4 = \pi - \theta_S - \theta_R > 0$ for each edge of the surface, i.e. $\theta_S + \theta_R < \pi$, or in other words, it measures the distance of the receiver to the specular boundary $\pi - \theta_S$. This is illustrated in figure 5.14.

Once a reflecting surface has been found, it is possible to locate the reflecting point on the surface. This method involves deriving the *barycentric* coordinates of the point within the reflecting (triangular) face. This method is similar to a method used in computer graphics to find line-triangle intersections.

To find the coordinates of the reflection point, it is helpful to consider first a 2D case, as in figure 5.15, to get an expression for the distance x of the reflection point to the edge, in terms of the diffraction parameters r_S, r_R, θ_S

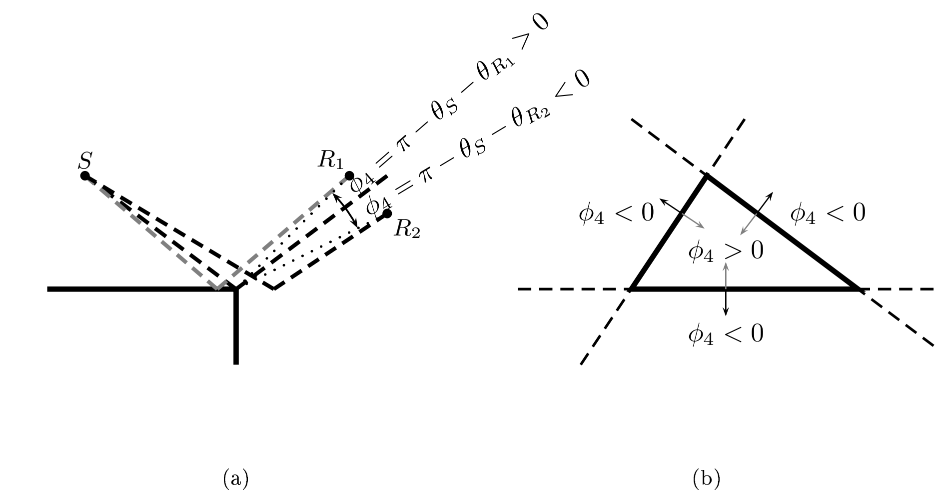


Figure 5.14: Checking for specular reflections. (a) the face tested is shown as a horizontal line, and the edge comes out of the page. The diffraction parameter ϕ_4 from equation (5.6) measures the angular distance of the receiver from the specular boundary. If $\phi_4 > 0$, for example for R_1, the sound is reflected by the face to the receiver. (b) If $\phi_4 > 0$ for all three edges of the face, the face creates a specular reflection.

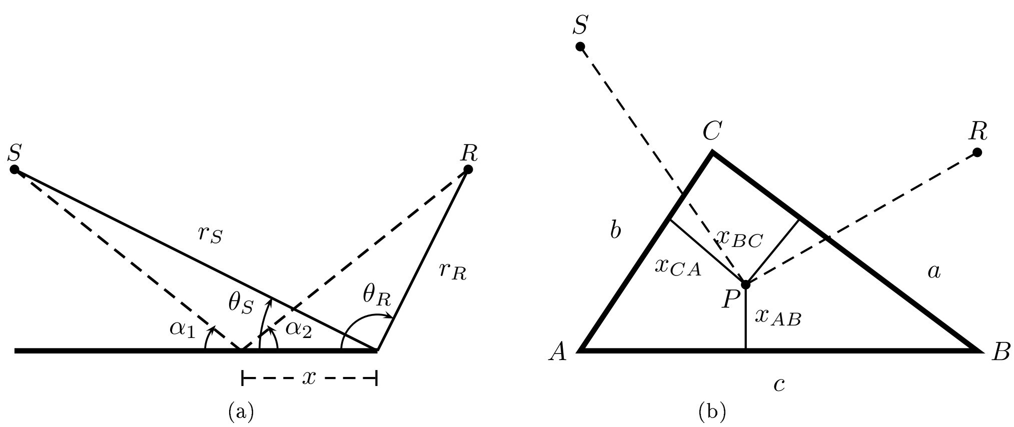


Figure 5.15: Finding the reflection point P. (a) 2D geometry to find the distance x of P from an edge. (b) The values x_{AB}, x_{CA}, x_{BC}, each calculated with equation (5.45) relative to the three edges give the exact position of P in trilinear coordinates, which can be converted to barycentric coordinates (equation (5.46)), that can be used to calculate the Cartesian coordinates (equation (5.48)).

and θ_R. When $\alpha_1 = \alpha_2$ (true for a specular reflection), x is given by

$$x = \frac{r_S r_R \sin(\theta_S + \theta_R)}{r_S \sin\theta_S + r_R \sin\theta_R} \tag{5.45}$$

For the 3D case, consider a source S, receiver R, and a triangular face $\triangle ABC$, as illustrated in figure 5.15, for which $\phi_4 > 0$ for all three edges. The first order specular reflection path from S to R must go through a point P in the interior of the face; the value x in equation (5.45) corresponds to the perpendicular distance from P to the edge for which the diffraction parameters are measured. Thus we can obtain a triple (x_{BC}, x_{CA}, x_{AB}) representing the location P in exact trilinear coordinates. This triple can be converted to barycentric coordinates (t_1, t_2, t_3) [Wei05b], where

$$t_1 = \frac{x_{BC}a}{n}, t_2 = \frac{x_{CA}b}{n}, t_3 = \frac{x_{ABC}c}{n} \tag{5.46}$$

with a, b and c the lengths of the sides of the face, and

$$n = ax_{BC} + bx_{CA} + cx_{AB} \tag{5.47}$$

Using the known Cartesian coordinates of the triangle vertices A, B and C, the Cartesian coordinates of P can be found with

$$P = A \cdot t_1 + B \cdot t_2 + C \cdot t_3 \tag{5.48}$$

This is a similar representation as the parametric coordinates of a point P in a plane[4]:

$$P = V(s,t) = A + s\vec{u} + t\vec{v} = (1 - s - t)A + sB + tC \tag{5.49}$$

using the direction vectors $\vec{u} = B - A$ and $\vec{v} = C - A$, and where s and t are real numbers, which are coordinates for the plane relative to the origin A and the basis vectors $\vec{u}$ and $\vec{v}$. When $0 <= s$, $0 <= t$, and $s + t <= 1$, $P = V(s,t)$ is inside the triangle $T = V_0V_1V_2$.

In our source object model, specular reflections can only occur in the case of a non-convex geometry, i.e. when the object has indents. The reflective properties of the surface can be taken into account, by adding an extra filtering function $R(\omega)$ to the reflected sound. The reflected sound will also be diffracted on the edges of the indent, that are not part of the reflecting face.

[4] see `http://www.softsurfer.com/Archive/algorithm_0104/`

Chapter 6

Implementation: *swonder3Dq*

Software was written to implement the calculations for the GNU/Linux platform, as an open source project under the GPL-license. The software is an extension of the *sWONDER*-software, though it has initially been developed as a separate project. In this chapter several aspects of the implementation will be discussed in detail, as well as aspects for the final integration with the software described in chapter 3.

6.1 Overview of software components

The software is (currently) divided into three executable programs and a library containing the common classes and data structures.

Graphical user interface

One of the programs is a graphical user interface (GUI), created using the *Qt-libraries* [Tro05]. This GUI (see figures 6.1 and 6.2) enables the user to define a project containing several objects and calculate the loudspeaker filters for WFS reproduction.

The objects themselves are defined by their geometrical data and the radiation filters at several points on the surface. Objects can be positioned and given different orientations in space. The user can define filters for each node, and the program calculates loudspeaker impulse response filters based on the WFS operators and saves them to disk.

Renderer

A second part of the software was written to enable scriptable calculations, based on a project. This program takes a script with commands to change diffraction order of an object, or a location, or to refine an object, and then execute the WFS calculation for specific objects. The program outputs a log-file in which is described which commands were executed, where the results of the calculation are stored and how long each command took to execute. A summary of script

Figure 6.1: Snapshot of the main window of the graphical user interface of *swonder3Dq*.

commands is given in table 6.1.

ENGINE

A third part of the software enables the user to load a project and listen to the desired object on the desired location with the desired orientation. This part of the software can be controlled with OpenSoundControl (OSC) [WFM03]. An overview of the OSC commands is given in table 6.2. *BruteFIR* [Tor05] is used as the convolution engine. There is a simultaneous visualisation of the scene possible using *GeomView* (see below).

6.2 Internal data structures

The internal data structure has at its root a Project, which contains the speaker Array, a list of Objects and a list of Scenes. Each element of the structure is implemented as a C++-class. In figure 6.3 there is a general overview of the data structure.

The `Array` refers to the loudspeaker array used for the WFS reproduction. It contains two lists, the first being a list of `ArraySegments`, the second a list of `ArraySpeakers`. `ArraySegments` are used to define linear parts of the array, from which the actual loudspeaker positions are calculated with the function `Array::calcspeakerpos()`.

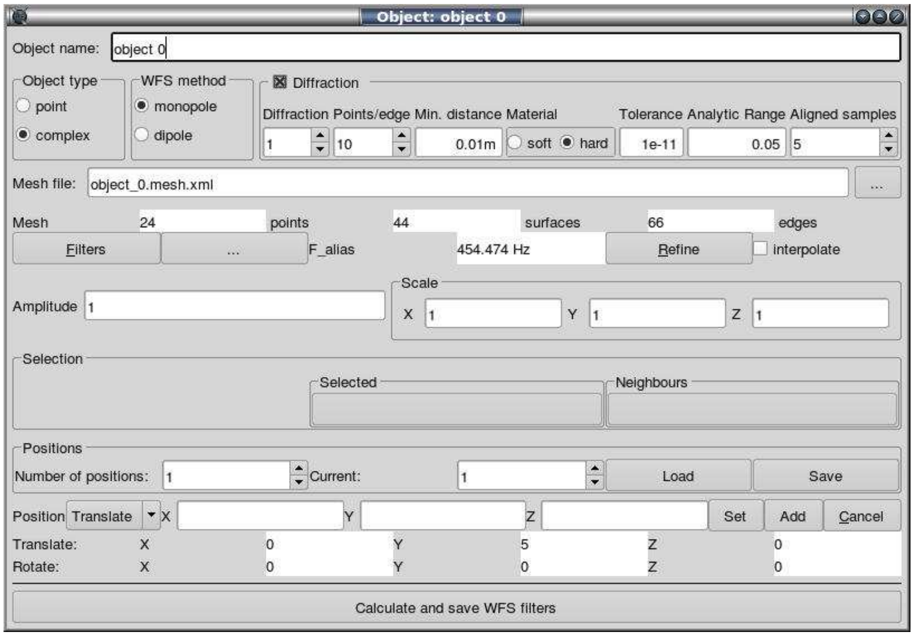


Figure 6.2: Snapshot of the object definition window of the graphical user interface of *swonder3Dq*.

The `Object` refers to the definition of the sounding object. The object has a geometrical definition in the form of a triangulated `Mesh`, has scaling factors `scale` and can have a number of positions as defined by the list of `ObjectCoordinates`. The scaling factor allows for a scaling of the `Mesh` in each of the three dimensions and is fixed for each object.

The list of `ObjectCoordinates` describe coordinates for which we want to calculate the WFS response of the loudspeakers. The object coordinates consist of a translation vector, indicating the position of the object, and a rotation vector, indicating the rotation of the object around each axis. The results of the WFS calculation are stored in a binary file: one for each location of each object. This file contains all the impulse responses for all the speakers defined in the speaker array. The class `CoefSet` contains a description needed to use the data for convolution, and contains information about the speaker id to which the coefficients apply, the data format (e.g. Big Endian floats), the block size, the offset and the name of the file. With this information the *engine* can generate a BruteFIR configuration file.

The `WFS_Mesh`-class inherits from the `Mesh`-class, which contains only the geometric description, and contains the complete definition of the object. The basic geometry is divided into `Vertices`, `Edges` and `Triangles`, where each `Edge` is made up of two `Vertices`, and each `Triangle` is made up of three `Edges` and three `Vertices`. These references are stored internally using pointers.

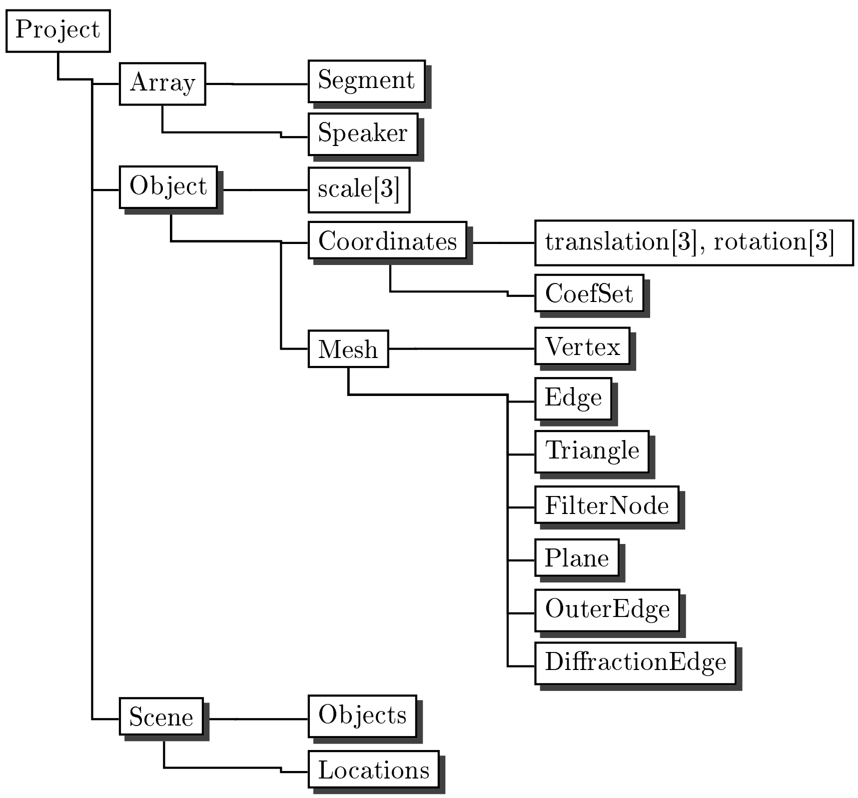


Figure 6.3: Overview of the data structure within *swonder3Dq*. A shadowed box means there is a list of the named object within the parent class. Each element is implemented as a C++-class (except scale, translation and rotation). A description is given in section 6.2.

command	*arguments*	*description*
`refine`	ijb	refine object i j times with or without interpolation of the filters
`diffraction`	ij	set the order of diffraction of object i to j
`addlocations`	ij	add j locations to object i
`translate`	$ijxyz$	translate location j of object i with (x, y, z)
`rotate`	$ijxyz$	rotate location j of object i with (x, y, z)
`scale`	$ijxyz$	scale location j of object i with (x, y, z)
`calculatewfs`	i	calculate wfs filters for object i. If no argument is given, calculate for all objects
`#`		indicates a comment. The text of the comment is copied into the log file

Table 6.1: Available script commands.

Additionally each `Vertex`, `Edge` and `Triangle` has a unique name, by which it can be identified. The acoustic properties for each `Vertex` are stored in a `FilterNode`, which contains a pointer to the `Vertex` it belongs to. The filter definition is described in more detail in section 6.4.

The `WFS_Mesh` class has a method, `WFS_Mesh::find_planes_and_outeredges`, to find from the geometric description how many different `Planes` are within the object, and the `OuterEdges` of each `Plane` (see section 6.3). These are then used to create the `DiffractionEdges` with `WFS_Mesh::create_diffraction_edges`. This will be described in more detail in section 6.6.

A `Scene` contains two coupled lists: the first is a list of `Object` id's which are part of the scene, and the second is a list of the location id's of where these objects are in the scene. Creating `Scenes` in a `Project` allows to calculate the resulting WFS speaker response for a constellation of sources, including the diffraction of the sound of one source around another object (by setting all of the filters of an object to zero, non-sounding objects can be created, around which sound from other sources is diffracted)[1].

6.2.1 File format

All data is stored in an XML file format, which is handled by a set of classes: `XML_WFS`, `XML_Project`, `XML_Mesh` and `XML_Array`. The last three create files for the `Project`, containing information on all objects and scenes, the `WFS_Mesh`, and the `Array`. The class `XML_WFS` contains functions which create the actual XML data structure. For the XML implementation, the library *libxml++-2.6*[2] is used, which is available in Linux distributions.

[1]The scene calculation has not actually been implemented yet, though the basic infrastructure (as described here) is there.

[2]`http://libxmlplusplus.sourceforge.net/`

Messages the engine understands:

command	*arguments*	*description*
`/start`		start the engine
`/stop`		stop the engine
`/verbose`		print messages to stdout
`/quit`		quit the program
`/project`	s	load project file
`/change`	ii	move object i to location j
`/info`	s `render`	get info on render status
`/info`	s `project`	get info about project
`/info`	si `object`	get info about object i
`/info`	sii `location`	get info about location j of object i
`/client`	ss	set client address and port for feedback
`/mute`	is	mute object i on or off
`/geomview/start`		start the viewer
`/geomview/stop`		stop the viewer
`/geomview/array`		show the speaker array
`/geomview/project`		show the project
`/geomview/top`		open top view
`/geomview/front`		open front view

Messages the engine sends:

command	*arguments*	*description*
`/swonder3d`	s	general feedback
`/swonder3d/change`	ii	object i moved to location j
`/swonder3d/mute`	is	object i mute in state j
`/swonder3d/render`	s	engine `running` or `not running`
`/swonder3d/project`	si	project with name i and j amount of objects
`/swonder3d/object`	issii	object i with name j, mesh file k, l amount of objects, currently at location m
`/swonder3d/location`	siiffffffffff	object with name i and id j, location k, with xyz-parameters for translation, rotation and scale

Table 6.2: OpenSoundControl namespace. In the arguments column “s” means string, “i” means integer and “f” means float. In the description where necessary the arguments are referred to with italic letters in alphabetical order, starting with i, reflecting the order in which the arguments are passed on with the message.

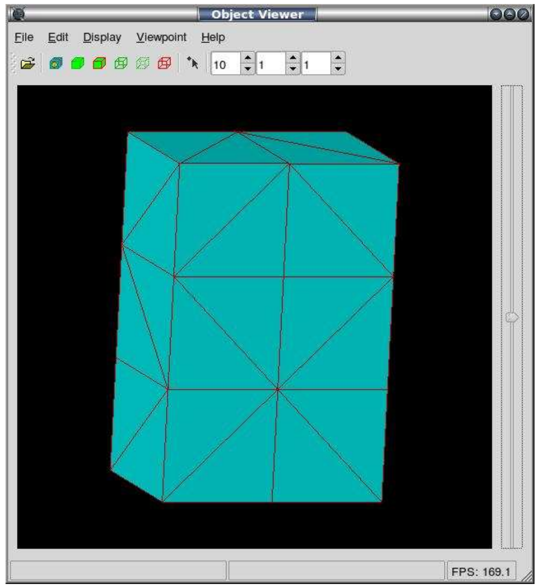


Figure 6.4: Graphical display of an object with *mview*.

6.3 Mesh

6.3.1 Choice of viewers and library

There are a multitude of programs and libraries available for manipulating and visualising 3D data. The *GTS-library* [gts05] is a C-library and encompasses many functions to read, write and work with meshes. A disadvantage of this library is that the points on the mesh (the vertices) are not ordered or tagged while they are loaded, so it is not easily possible to connect the data for the radiation filters to them.

In the C++-program *mview* [Can05] a lot of methods for working with meshes had already been implemented and with only a few additions to implement the filter definition per source point, it is incorporated into *swonder3Dq* for the graphical display of the objects (see figure 6.4).

GeomView [Tec05] is used as a second viewer to view the whole scene: the WFS speaker array as well as several sounding objects (see figure 6.5). *GeomView* is used as an external program, interfaced via a (file) node.

6.3.2 Supported file formats

As the basic mesh code is borrowed from *mview*, *swonder3Dq* can read the following mesh formats:

- PMesh format (used at the Vision group of the University of Edinburgh),
- GTS format (from the Gnu Triangulation Library),

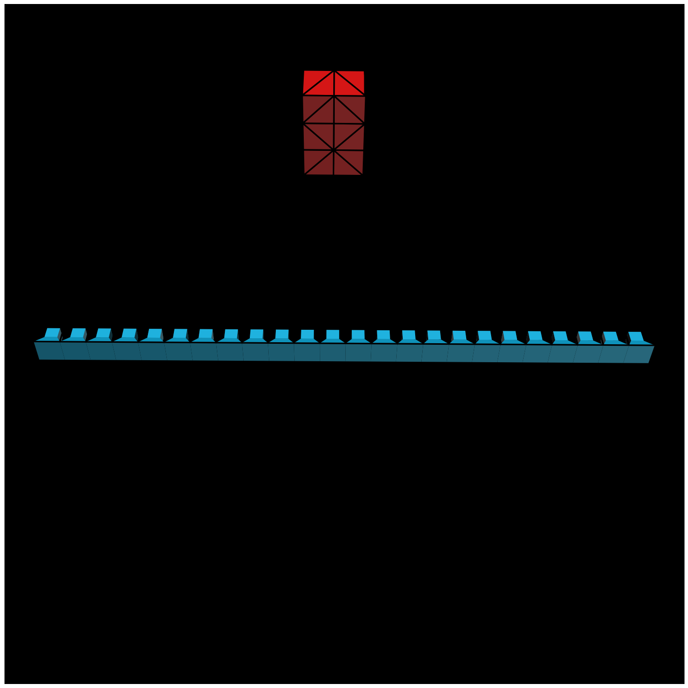


Figure 6.5: Visualisation of a WFS-Scene with *GeomView*.

- Geomview format (only format "OFF" or "COFF"),
- PLY format (only ASCII format),
- VRML 1.0 format,
- VRML 2.0 format (VRML97),
- Visualisation Toolkit VTK format (ASCII POLYDATA only),
- Alias Wavefront / Java 3D OBJ format (ASCII polygon data only).

The mesh must be a triangulated mesh for the calculations within *swonder3Dq*.

In order to display a scene *swonder3Dq* stores the mesh in the GeomView format, so that GeomView can load and display the mesh.

As mentioned above, *swonder3Dq* uses its own XML-format to store the mesh. This is due to the fact that it needs to store much more information than can be contained in the other formats.

6.3.3 Refinement of the mesh

To make the discretisation distance smaller, an algorithm is needed to calculate more points on the object surface. A simple method (inspired by a method in the *GTS Library*) is the mid vertex insertion method. On each edge of a triangle, a point is added in the middle to divide the edge in two. Then, every

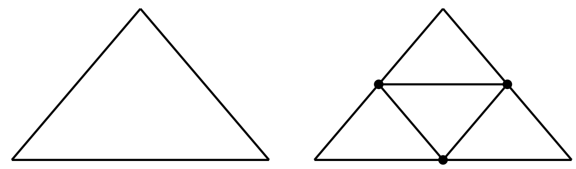


Figure 6.6: Refinement of a triangle with the *midvertex insertion* method

midpoint is connected to those of the other edges (figure 6.6). This method can be applied multiple times to create a fine raster of points. An estimate for the aliasing frequency (see section 5.3) can be calculated by finding the longest edge in the mesh and calculate the corresponding aliasing frequency from equation (5.2).

Additionally the filter for the newly calculated points needs to be determined. This is done by an average of the filter of the neighbouring points, using an inverse distance weighting method [Gre00] to determine the contribution of each point:

$$Z_j = \frac{\sum_{i=1}^{n} \frac{Z_i}{h_{ij}^{\beta}}}{\sum_{i=1}^{n} \frac{1}{h_{ij}^{\beta}}} \tag{6.1}$$

Z_j is the value of the new point j, Z_i of the neighbour point i, h_{ij} the distance from point i to j, and β a factor that defines the weighting of the distance, usually set to $\beta = 2$. In this case the factor β can be interpreted as a kind of measure how well the sound is propagated through the material of the object.

Further aspects of the filter averaging will be discussed in the next section.

6.3.4 Finding planes and outer edges

Algorithm C.2 finds the planes and outer edges in the mesh, which will be used for the diffraction calculation. It calls the function `WFS_Mesh::number_planes()` (algorithm C.1), which iterates over all the triangles, assigns a plane numbers to the triangle (if it does not already have one) and gives any other triangles in the same plane the same number. The check for the same plane is a check whether the normals of the triangles have the same direction, and whether the distance of the surface to the origin is the same (see the function `Triangle::on_same_plane` in the source code).

When a new outer edge is created, it automatically looks for any other edges that are on the same outer edge. As each `Edge` contains references to the two `Vertices` it is made of, it can check whether other `Edges`, of which that `Vertex` is a part, are on the same line. This can be done iteratively, until no more `Edges` are found.

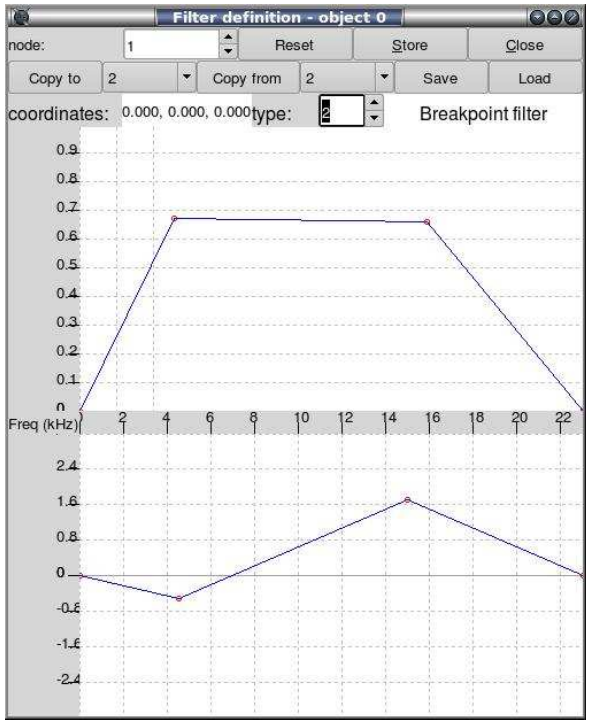


Figure 6.7: Snapshot of the dialog to define a filter for a point on the source object surface.

6.4 Filter definition and calculation

For each vertex of the object a function $G(\vec{r}, \omega)$ needs to be defined. There are five options for this function:

1. No filter, i.e. $G(\vec{r}_N, \omega) = 0$.
2. Unity filter, i.e. $G(\vec{r}_N, \omega) = 1$. There is an amplitude parameter to set the actual response to another sound level.
3. A breakpoint filter, i.e. the function $G(\vec{r}, \omega)$ is defined by a number of breakpoints for frequency in magnitude and phase.
4. An impulse response from file.
5. A filter defined in the frequency domain (this format is used internally).

There is a simple graphical representation of the frequency response of the filter, where the user can define breakpoints (figure 6.7). The filter settings can be copied between source points and there is an interaction between the picked triangle and its corner points (which can be selected in the GUI) and the current source point for which a filter is defined.

6.4.1 Filter response of the breakpoint filter

In the case of a breakpoint filter, the filter's response is calculated with a method as described in [Smi97, ch. 17, p. 297-300]. The following steps are taken in the calculation:

1. Convert the breakpoints to actual bin values for the frequency magnitude and phase response. This is done in the function `FilterNode::cornerPointsToBins`. The method has as a constraint that the first and last sample must be 0.
2. Convert from a magnitude/phase to a real/imaginary representation.
3. The inverse discrete Fourier transform (DFT) is taken to move the filter into the time domain (see `Convolution::filterNodeToFFTW`). For the fast Fourier transform the *FFTW Library* is used [FJ05]:

   ```
   fftw_execute( backward_transform );
   ```

4. The time response is shifted with M/2 samples, where M is called the filter kernel:

   ```
   for ( int i=0; i < kernelsize/2; i++ )
   {
     temp[i][0] = filt[i + windowsize - kernelsize/2][0];
     temp[i][1] = filt[i + windowsize - kernelsize/2][1];
   }
   for ( int i=0; i < (windowsize - kernelsize/2); i++ )
   {
     temp[i+kernelsize/2][0] = filt[i][0];
     temp[i+kernelsize/2][1] = filt[i][1];
   }
   ```

5. The time response is windowed:

   ```
   for ( int i=0; i < kernelsize; i++ )
   {
     filt[i][0] = filt[i][0] * (0.54-0.46*cos( 2*M_PI*i/kernelsize ) );
     filt[i][1] = filt[i][1] * (0.54-0.46*cos( 2*M_PI*i/kernelsize ) );
   }
   ```

6. The time response is truncated:

   ```
   for ( int i=kernelsize; i < windowsize; i++ )
   {
     filt[i][0] = 0;
     filt[i][1] = 0;
   }
   ```

7. The time response is transformed back into the frequency domain:

   ```
   fftw_execute( forward_transform );
   ```

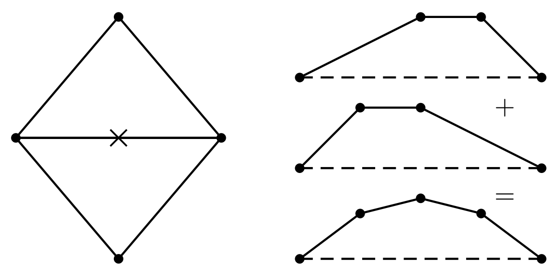


Figure 6.8: Averaging of the filter values. On the left is shown from which point the average is taken: $\times$ is the new point, $\bullet$ are the neighbourpoints. On the right is shown how the breakpoints are added when averaging.

6.4.2 Averaging filters

The weighted average between two filters is calculated as follows (see `FilterNode::weighted_average`):

- If both filters have no filter, the result has no filter (`type0`).
- If one of the filters has no filter, then the amplitude of the other filter is weighted (`type0`).
- If both filters have a unity response, the result has a unity response (`type1`).
- If both filters are of a breakpoint type: for each breakpoint from either filter the corresponding value on that frequency value is calculated for the other filter. The new filter then has a breakpoint value at that frequency value, which is an average of the two breakpoints of the two filters (figure 6.8). The average is taken from the real and imaginary parts of the coefficients (`type2`).
- If both filters are impulse responses, the average is taken of these (`type3`).
- If one filter has a unity response, and the other filter an impulse response, then the impulse responses are averaged (`type3`).
- In all other cases, both filters are converted to an FFT format (if they not already are in that format) and the average is taken of these. The result is kept in the FFT format (`type4`).

6.5 3D WFS Calculation

There is a choice between defining the source points as monopoles or as dipoles.

In the case of a dipole, the main axis of radiation is in the direction of the normal (pointing outwards) on the surface; in the case of points on the corners of triangles which are not on the same plane, this normal is an average of the normals of the triangles it is a corner point of.

Algorithm C.3 gives an overview of the calculation of the 3D WFS operator for the monopole source type; the calculation for a dipole source type follows

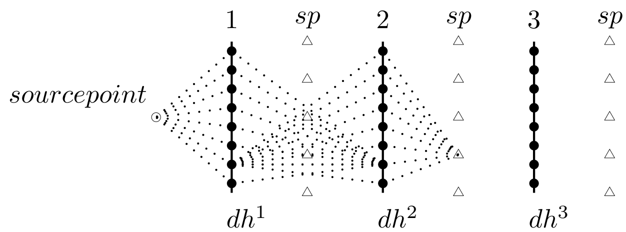


Figure 6.9: Illustration of the diffraction algorithm. *sp* indicates the speakers. For each 2nd order edge source the contributions dh^1 can be calculated and stored as an impulse response. For the final impulse response for a certain loudspeaker, then the contributions dh^2 can be calculated for each edge source, convolved with the impulse response of the previous diffraction, before adding all contributions up for the loudspeaker impulse response.

a similar scheme. Note: the factor $\sqrt{\frac{jk}{2\pi}}$ from equation (2.30) is not taken into account during the calculation; given an appropriate FIR version this filter it would be straightforward to add a convolution with this filter during the computation.

When a source point is at the back side of the object, a direct path of the sound to the loudspeaker is not possible and this source point should not be taken into account. The algorithm for this was described in the previous chapter. Diffraction of sound waves around the object is taken into account using the diffraction model from chapter 5 and implementation details are below.

Algorithms C.4 and C.5 give an overview of the complete procedure of calculating the WFS speaker responses for an object.

The amplitude factor with which the speaker filter is multiplied before it is saved to disk consists of the amplitude that has been set by the user, and an amplitude correction for the number of points N on the mesh. The latter factor is

$$A = \frac{1}{\sqrt{N}} \tag{6.2}$$

which can be easily derived, as the sound pressure level is dependent on the square of the amplitude of the wave.

6.6 Diffraction Model

In section 6.3 was described how the planes and outer edges of the object are found. Then (see algorithm C.6) for each source point emitting sound, first order diffraction edges are determined. After that, the higher order diffraction edges are determined. Each diffraction edge holds a reference to its source function (the filter of the source point), its previous order diffraction edges, and its higher order diffraction edges. Based on this information, a diffraction model of the object can be calculated.

The next step is to prepare the diffraction impulse responses, in as far as possible. To calculate the impulse response of a diffraction edge, we need to know the source position, the secondary (or edge) sources, and the receiver position. As in our case the source position is fixed (with respect to the object), we can calculate the (multiple) diffraction impulse responses up to the secondary sources of the last edge, which act as receivers for the second last edge (see figure 6.9 and equation (5.19)). Thus, these intermediary impulse responses can be calculated for the object (this is done in the function `WFS_Mesh::prepare_diffraction_irs`, described in algorithm C.7), and only for the final impulse response, we need to calculate the contribution from the last edge source to the receiving point (in our case the loudspeaker), convolve that with the impulse response of previous orders for that edge source, add up the contributions from all edge sources, and save it as the impulse response for the speaker.

In algorithm C.6, it can be seen that an argument called *min. distance* is used. This argument sets the minimum distance two edge elements need to have, before multiple diffraction is taking place between them. The minimum distance given determines from what distance on the contribution is windowed. If this is not done, then, especially when high orders of diffraction are calculated, multiple diffraction between nearby edge elements contribute with each order, and as the travel time differs only marginally from previous order contributions, they add up (too) rapidly[3].

In the function `DiffractionEdge::calculate_edgesource_response` (algorithm C.8), the response for an edge is calculated. This is done by first finding the sample aligned sources, after which is checked within which edge segments they are. The user can set how many samples need to be aligned for the first order; for higher orders always only the first sample is aligned to ensure that the onset sample is calculated correctly. Then the contribution for each sample aligned source is calculated, and, for the first order, placed into the result vector. For higher order diffraction, the result is placed into the vector, when the response for the edge segment within which the sample aligned source lies is calculated.

Finally, the contributions for all edge segments are calculated. There needs to be iterated twice over the edge segments, as in the second run we place the contributions into the result vector and in order to apply slope correction, we need to know the contributions of all edge segments.

6.7 Summary of the calculation for each object

Summarised, the calculation method for each object is:

1. Calculate the diffraction model.
    (a) Find the edges of each plane surface.
    (b) Define the diffraction edges.

[3] For such small distances it cannot be assumed that the diffraction theory is valid.

   (c) Calculate the intermediate impulse responses.

2. For each location:
   (a) Transform the mesh to the new location.
   (b) Calculate the WFS delay and attenuation (for each speaker) for each source point and each secondary source point, this includes a visibility check.
   (c) Calculate the resulting impulse response for all diffraction edges for each speaker position as a receiver.
   (d) Convolve all of the impulse responses with the source filter.
   (e) Save the filters to disk.

6.8 Integration with *sWONDER*

In chapter 3 a general structure for a component based WFS software was presented. The software *swonder3Dq* can be integrated into this system with the following modifications:

- The addition of a *3dqrender* stream to *cwonder*, which takes care of the communication between a user interface and a non-realtime renderer for the calculation of the impulse responses.
- Realtime rendering using *fwonder*, instead of BruteFIR.
- Adaptation of *swonder3Dq* to split the offline rendering from the GUI, so there will be an OSC-controlled offline renderer and a GUI that communicates via *cwonder* with the offline renderer and the realtime renderer.

Additional OSC commands for the communication are shown in table 6.3.

6.8.1 Realtime control

The realtime rendering can be done with *fwonder* as a convolution engine, which means that during the non-realtime rendering the file format for impulse response storage must be adapted.

cwonder needs to be extended to be able to select the correct impulse responses for *fwonder* to load.

First of all the command `/WONDER/source/type` needs to be interpreted such that type 3[4] means, that a 3D object is chosen. With a further command `/WONDER/source/object` the actual object can be selected, upon which *cwonder* can send back a message to the visual stream with the mesh data of the stream (`/WONDER/source/mesh`).

If a source becomes a 3D object, *twonder* must mute (or fade out) that source, while *fwonder* must unmute (or fade in) that source, as well as load the appropriate impulse responses.

As 3D objects can be rotated, the command `/WONDER/source/rotation` needs to be added.

[4] type 2 is reserved for multipoles

command	types	arguments
/WONDER/source/rotation	iffff f	src id, rot x, rot y, rot z, time, duration
/WONDER/source/object	is	src id, name
/WONDER/source/mesh	isfffb	src id, name, scale x, y, z, mesh data
/WONDER/3dqrender/object/add	isb	object id, name, mesh data
/WONDER/3dqrender/object/point	i	object id
/WONDER/3dqrender/object/mesh	ib	object id, mesh data
/WONDER/3dqrender/object/scale	ifff	object id, scale x, scale y, scale z
/WONDER/3dqrender/object/filternode	iiib	object id, vertex id, type, data
/WONDER/3dqrender/object/diffraction	iiffff	object id, diffraction order + parameters
/WONDER/3dqrender/object/refine	ii	object id, times of refinement
/WONDER/3dqrender/location/add	ii	object id, location id
/WONDER/3dqrender/location/translate	iifff	object id, location id, translation x, translation y, translation z
/WONDER/3dqrender/location/rotate	iifff	object id, location id, rotation x, rotation y, rotation z
/WONDER/3dqrender/scene/add	i	scene id
/WONDER/3dqrender/scene/delete	i	scene id
/WONDER/3dqrender/scene/object/add	iii	scene id, object id, location id
/WONDER/3dqrender/scene/object/delete	iii	scene id, object id, location id
/WONDER/3dqrender/calculatewfs	i	object id
/WONDER/3dqrender/object/calculatewfs	i	object id
/WONDER/3dqrender/location/calculatewfs	i	object id, location
/WONDER/3dqrender/scene/calculatewfs	i	scene id
/WONDER/3dqrender/task/done	i	task id
/WONDER/3dqrender/task/error	iis	task id, error id, error message
/WONDER/3dqrender/task/progress	iii	task id, progress, total
/WONDER/3dqrender/task/new	is	task id, command name
/WONDER/3dqrender/task/subtasks	ii	task id, number of subtasks
/WONDER/3dqrender/subtask/uniqueID	iii	parent task id, sub task id, unique id
/WONDER/3dqrender/task/stop	i	task id

Table 6.3: Additional OSC commands for integration of *swonder3Dq* with *sWONDER*.

When there are several sources (including 3D objects) at several locations chosen, *cwonder* needs to check whether a special scene calculation has been made for that constellation; if not, it can look for a simple superposition of the locations, if the chosen locations for the objects are available.

Furthermore, for realtime interaction with a user interface it will be useful to be able to request information about the available objects in the project, and their locations, as well as the available scenes. So this information needs to be passed on between *cwonder* and the user interface.

6.8.2 Non-realtime rendering

The offline calculation can benefit from the cluster architecture, as the calculations are for a large part only dependent on the receivers, i.e. on the WFS reproduction array speaker positions. Only the first part of the calculation, where the diffraction model is being defined, cannot be parallelised in this way. So for this part there are two possible solutions: (1) duplication of this part of the calculation for each cluster node, and (2) calculation of this part on the control node, after which the result is distributed to each other cluster node. The second option would be the best solution, but requires extra effort to implement.

As the cluster nodes have dual cores, with hyper-threading, further parallelisation can be useful to speed up calculations. Here we can benefit from the fact that calculations need to be done for each location of the object, and that these calculations are independent from each other. Thus we can create threads for each of these calculations, which are then processed independently.

This can be achieved by constructing the program having two main functions: the first is the OSC-server, which waits for messages from *cwonder* with commands for processing. For each command a task is created and subdivided. Information about the task, and how it has been subdivided is sent back to *cwonder*, with the message `/WONDER/3dqrender/task/new` to inform about the task id, and `/WONDER/3dqrender/task/subtasks` about the number of subtasks, and `/WONDER/3dqrender/subtask/uniqueID` about the numbers each of these subtasks have been labelled with. These tasks are put on a TaskQueue.

The TaskQueue is checked in regular time intervals, to check whether:

1. a task has been completed (`/WONDER/3dqrender/task/done`),
2. an error occurred while executing a task (`/WONDER/3dqrender/task/error`), upon which the task is stopped,
3. how far the task has progressed (`/WONDER/3dqrender/task/progress`),
4. a task was not running yet, and can be started now.

cwonder adds to each task message an extra argument, indicating the rendering unit, before passing the message on to the UI.

The proposed structure can be used for any other non-realtime rendering program.

With the `object`-commands the settings for objects can be set. The `add` command, adds a new object and sets the mesh data. If the object is in fact a point source (useful, in case it will become part of a *scene*), an extra message needs to be sent to set it to a point source (`point`). If the mesh data needs to be changed, or be renewed, the command `mesh` sets the data. With `scale` the object can be scaled. The acoustic properties for each vertex can be set with `filternode`, by indicating the vertex id, the type of filter, and data. Diffraction settings are set with `diffraction`. With the command `refine`, the object is refined N times.

With the `location`-commands, locations for the object can be added and modified.

With the `scene`-commands, constellations of sources can be created, for which the complete response will be calculated, i.e. the occlusion of sound sources by other objects is taken into account, as well as diffraction around these objects.

The `calculatewfs` commands start the actual calculation of all objects defined in the project, or for either an object, a scene, or a location of an object, with the corresponding subcommand.

Each of these commands creates a task; if no other calculations are running, most of them will be ready soon. Refinement of an object can take some time, if the object consists of a lot of triangles. The WFS calculations are also time consuming.

6.9 Future work

Though the basic algorithms have been implemented, there are some directions in which the implementation can be expanded, apart from the full integration with the other *sWONDER* components, as proposed in the previous section.

Future work should be directed towards:

- Optimisation of the calculation algorithm, making calculations faster.
- Including the calculation of audio scenes, where the sound of one source is diffracted around another object.
- Including local reflections on the object, in order to enable accurate reproduction of non-convex geometries (see subsection 5.6.2).
- Connecting the method with physical modelling of the source surface vibration, in order to provide an integrated method (see section 5.1)

Chapter 7

Evaluation

7.1 Physical effects

In the reproduction the following parameters can be varied: (1) shape of the object, (2) size of the object, (3) level of refinement, (4) diffraction (and many parameters of this, such as the order and number of segments), (5) filters for the points on the surface, (6) position of the object and (7) rotation of the object. Also the position of the listener herself has influence on the perceived effect.

To analyse the effect of the parameters in the reproduction, several signals were created and analysed. The signals were created for the reproduction setup as shown in figure 7.1.

We focus here on the effect of the shape, the size, level of refinement and the position of the object. The signals were calculated without diffraction, and a unity filter for each source point (so $G(\vec{r}_\Psi, \omega) = 1$ for all points).

7.1.1 Shape

Four different shapes were used: a point source, a sphere, a tetrahedron and an icosahedron (abbreviated as "icosa"), as shown in figure 7.2. The last three all had a diameter of $2m$.

The resulting speaker impulse responses for the test setup are shown in figure 7.3. Differences are clearly visible: the point source shows (as expected) a single wave front, whereas the more complex shapes show several wave fronts, from several directions. Depending on the refinement and the size of the object, the interference patterns between the wave fronts become more complex. The impulse responses are fairly short: up to 10 ms; this is in the range of direct and pseudo-direct sound. From the amplitudes we can see that depending on the shape, the amplitude at the reproduction speakers has a different distribution for each shape. Especially note how the contribution of the middle speaker tends to be the highest.

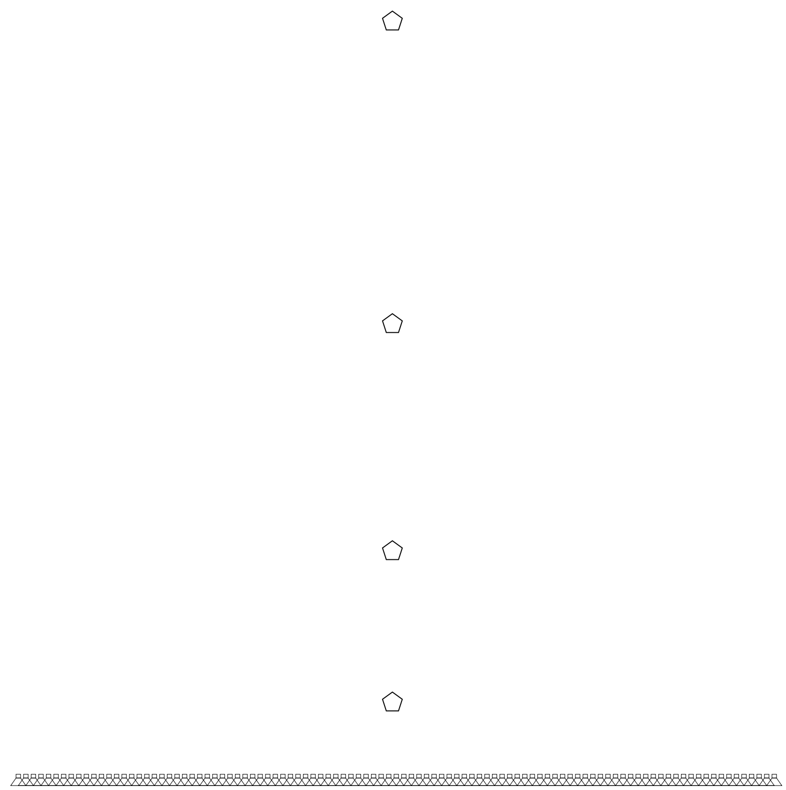
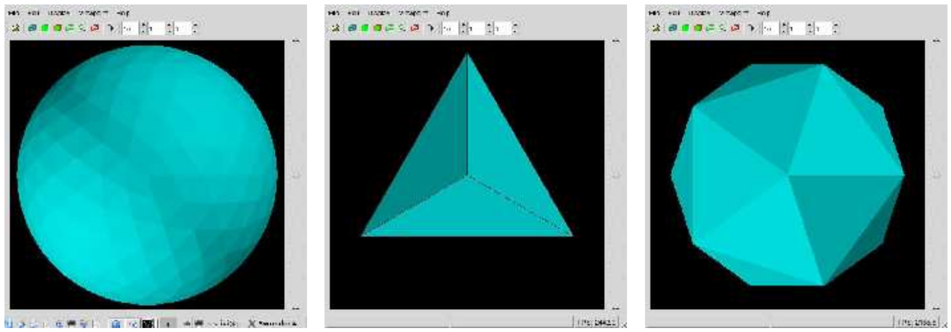
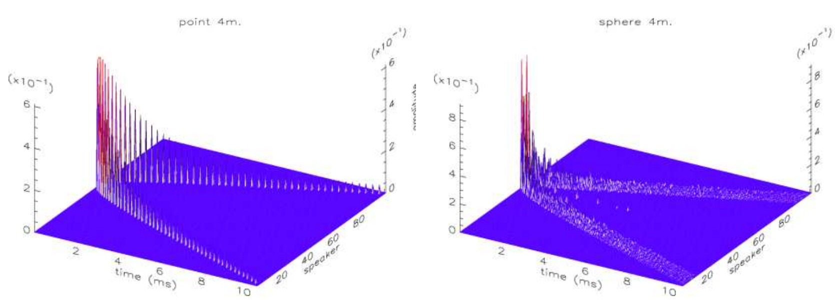


Figure 7.1: Virtual test setup for analysis of the physical effects. An array of 101 loudspeakers is used, with a total length of 10 meters. The pentagons indicate the virtual sound source position (1 m., 3 m., 6 m. and 10 m. behind the array).

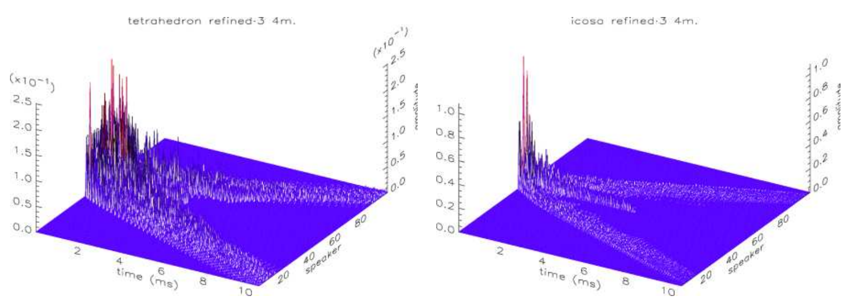


(a) sphere (b) tetrahedron (c) icosahedron

Figure 7.2: Different shapes used in the simulation and the listening test.



(a) point source (b) sphere



(c) tetrahedron (d) icosa

Figure 7.3: The loudspeaker impulse responses for different object shapes (at 1 meter behind the array) for the speaker setup used in the listening test (see figure 7.19). On the x-axis is the time in ms, the y-axis is the speaker number and the z-axis is the amplitude.

7.1.2 Size

Two objects (the tetrahedron and the icosa) were created with three different sizes: one the original size, one half the size, one quarter the size. For the tetrahedron the resulting speaker impulse responses for the test setup are shown in figure 7.4. Several effects can be distinguished: the wave form gets less wide with decreasing size, and the wave fronts get closer to each other. For the left-hand plots the decrease in size was done with the same number of points per surface element. This means that the discretisation distance was also decreasing with decreasing size. If the discretisation distance is held constant, the number of wave fronts is larger for larger objects, but the outer shapes are the same (i.e. the wave fronts from the most extreme points would still be there, and cause the same spatial extension; see the righthand plots of figure 7.4).

7.1.3 Refinement

Two objects (the tetrahedron and the icosa) were created with four different refinements.

For the tetrahedron the resulting speaker impulse responses for the test setup are shown in figure 7.5. With increasing refinement, the number of wave fronts interfering with each other increases. The wave fronts start to cancel each other out, which results in a smooth decay in the impulse response, which will cause a kind of lowpass filtering effect. This is shown on the right-hand side of figure 7.5, where the frequency responses are plotted. It can be clearly distinguished that with increasing refinement, the higher frequency components are more attenuated.

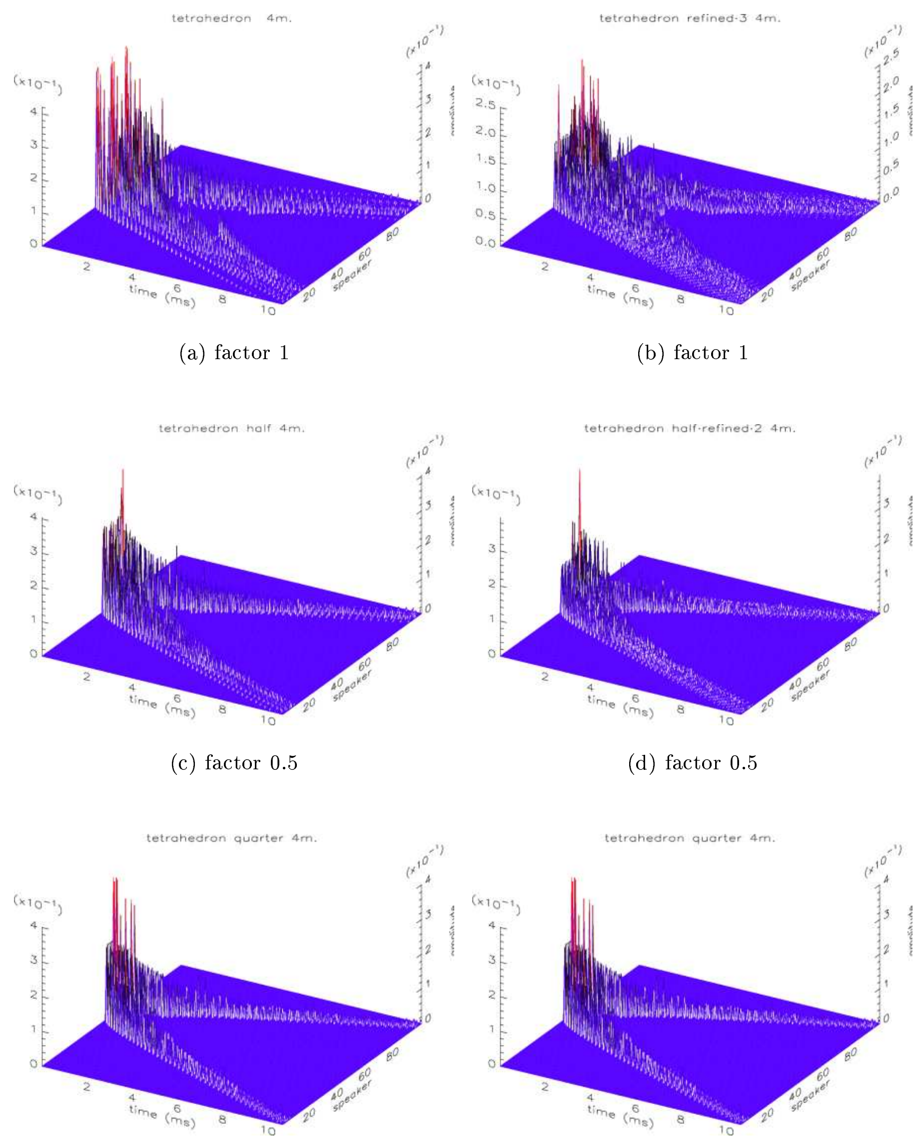
This can be understood if we look at the response of a line source (vibrating in phase) at a certain point in space (see figure 7.6). The response in point R is the integral of a point source response (see eq. 2.8) between point $\vec{r_1}$ and $\vec{r_2}$:

$$\int_{\vec{r_1}}^{\vec{r_2}} \frac{1}{2\pi r}\hat{p}(t-\frac{r}{c})\mathrm{d}r = \int_{\vec{r_1}}^{\vec{r_2}} \frac{1}{2\pi r}\delta(t-\frac{r}{c})\mathrm{d}r \tag{7.1}$$

So there will be a filtering of the sound, as compared to the response of a point source. The frequency will be dependent on the length of the line source (so there is a relation to the diameter of the source), and the number of points. This is illustrated in figure 7.7. With increasing length, we see that the impulse response get longer and the lowpass filtering gets stronger. With decreasing refinement, we see that the higher frequencies become less attenuated, but also more irregularly filtered.

7.1.4 Distance

All objects were reproduced at four different distances from the array: 1 meter, 3 meters, 6 meters, and 10 meters behind the array. For the tetrahedron shape the resulting speaker impulse responses for the test setup are shown in figure 7.8. It can be seen that the curvature of the wave fronts decreases with increasing distance, causing the speaker impulse responses to be more similar to each other.



(a) factor 1

(b) factor 1

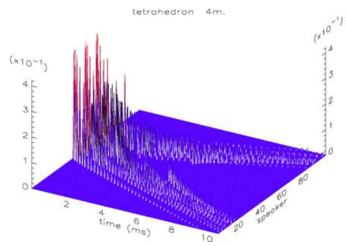
(c) factor 0.5

(d) factor 0.5

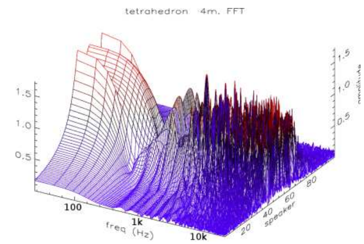
(e) factor 0.25

(f) factor 0.25

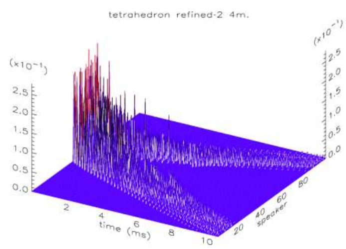
Figure 7.4: The loudspeaker impulse responses for the tetrahedron of different sizes for the virtual speaker setup (see figure 7.1). The left-hand side plots show the results with constant number of points on the surface, the right-hand side plots with constant discretisation distance.



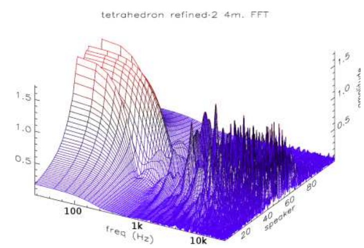
(a) 4 triangles



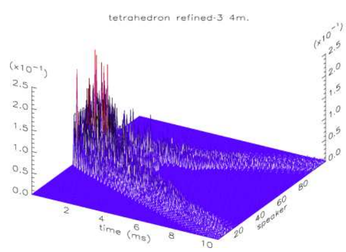
(b) 4 triangles FFT



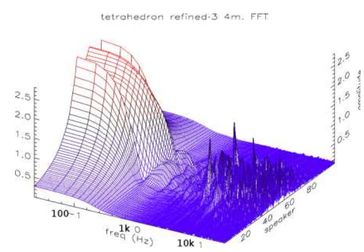
(c) 16 triangles



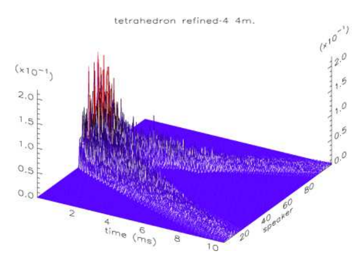
(d) 16 triangles FFT



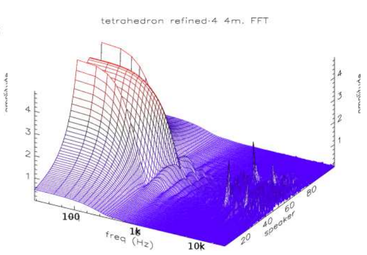
(e) 64 triangles



(f) 64 triangles FFT



(g) 256 triangles



(h) 256 triangles FFT

Figure 7.5: The loudspeaker impulse responses for different refinements of the tetrahedron for the virtual speaker setup (see figure 7.1). On the left-hand side the time domain plots, and on the right-hand side the frequency domain plots.

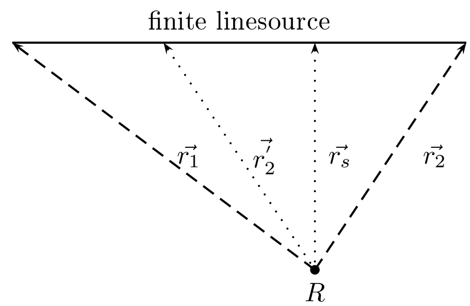


Figure 7.6: Calculation of the impulse response of the finite line source in receiver point R.

In the frequency domain it can be seen (on the right-hand side of figure 7.8) that the response is more similar for neighbouring speakers. Furthermore, there are some subtle differences in the frequency response. Especially note how at close distances there is a 'jump' in the response on the side (e.g. at $1m$ near the speakers 40 and 60). This jump is a result of not calculating diffraction, as from one speaker to the other, a source point on the object becomes 'visible' for that speaker. At larger distances, the angle between the speaker and the object becomes more or less the same for all speakers, and the 'jump' falls out of the scope of the speaker array.

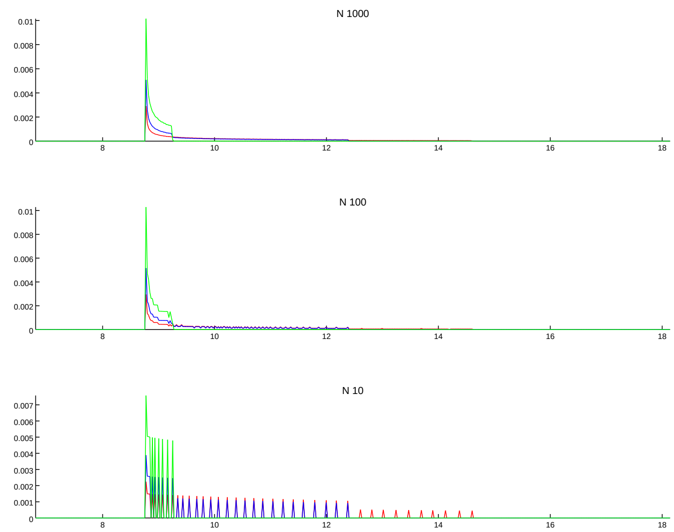
With increasing distance, the signals become slightly less filtered, which can again be explained by looking at the impulse response of a line source. With increasing distance, the resulting impulse response becomes shorter (but later in time), thus creating a flatter impulse response (see figure 7.9).

7.2 Diffraction effects

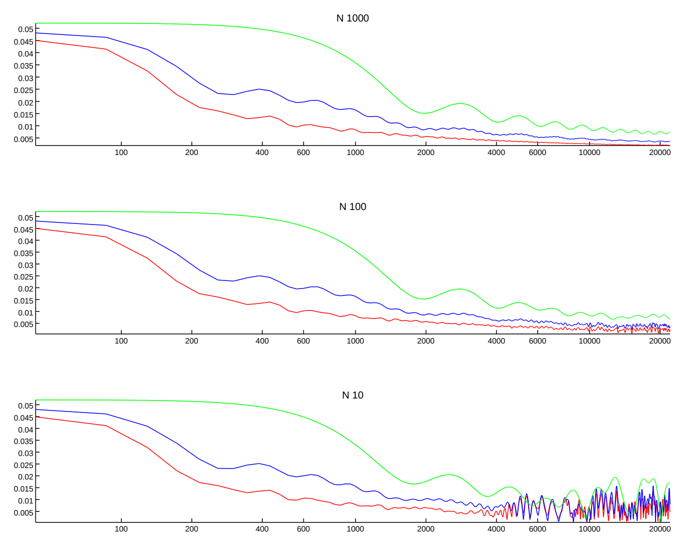
In order evaluate the diffraction model, an example was calculated for the scene shown in figure 7.10, modelled after the example discussed in [Van91]. The source object is a model of a box loudspeaker, with a front baffle dimension of 0.4m by 0.64m., having a box depth of 0.32m. A point source is placed at 0.2m from the top and side edges. The box is placed such that the point source is at the height of the loudspeaker array. The calculations were done for two distances of the object placement: 2 and 7 meters behind the loudspeaker array, and two rotations of the object: one with the face towards the speaker array, and one with the back to the array. The loudspeaker array consisted of 101 loudspeakers at 10cm distance from each other. A Hanning taper was applied with a width of 20%.

7.2.1 Loudspeaker signals

The loudspeaker signals that are calculated are shown in figures 7.11a (for the speaker model turned towards the array) and 7.11b (for the speaker model turned away from the array) for first, second and third order calculation. We

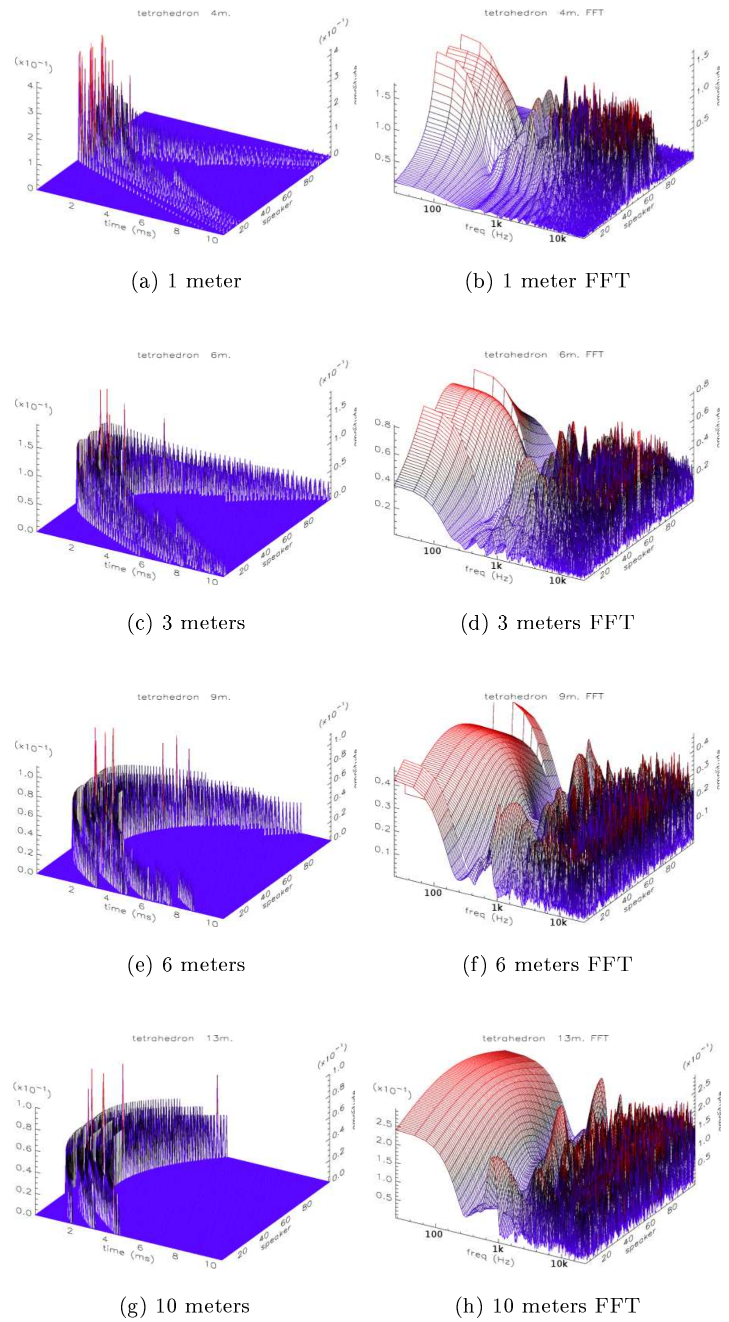


(a) time response



(b) frequency response

Figure 7.7: Impulse responses in the time (top) and frequency (bottom) domain of a simulation of a line source. The green line (upper) is for a line source of 2 meters long, the blue line (the middle) of 4 meters, and the red one of 7 meters (lower).



(a) 1 meter

(b) 1 meter FFT

(c) 3 meters

(d) 3 meters FFT

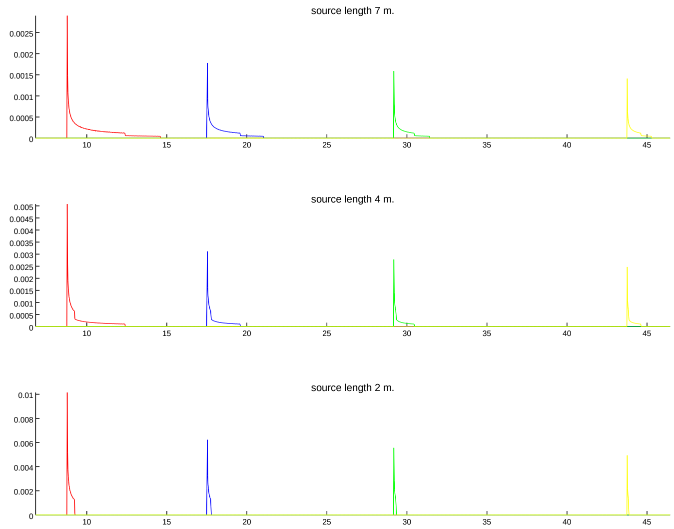
(e) 6 meters

(f) 6 meters FFT

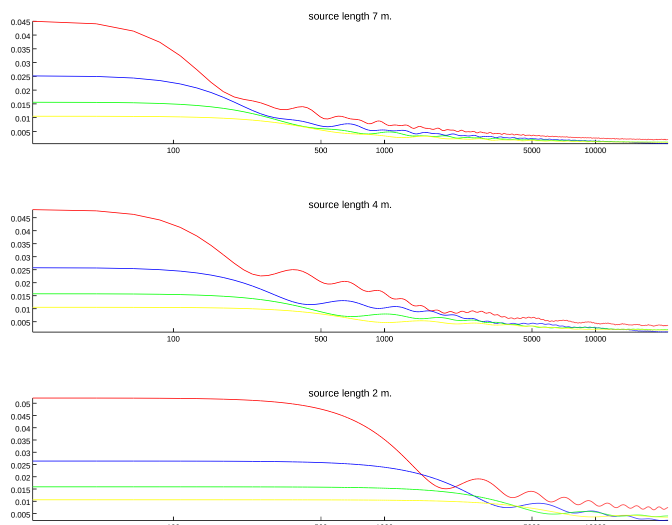
(g) 10 meters

(h) 10 meters FFT

Figure 7.8: The loudspeaker impulse responses for different distances of the tetrahedron for the virtual speaker setup (see figure 7.1).



(a) time response



(b) frequency

Figure 7.9: Time response (a) and frequency response (b) for a line source at different distances. The first (red) response is at 3 meters distance, the subsequent ones are at 6 meters (blue), 10 meters (green) and 15 meters (yellow) distance. We see that with increasing distance, the frequency response becomes flatter, and there is relatively less lowpass filtering.

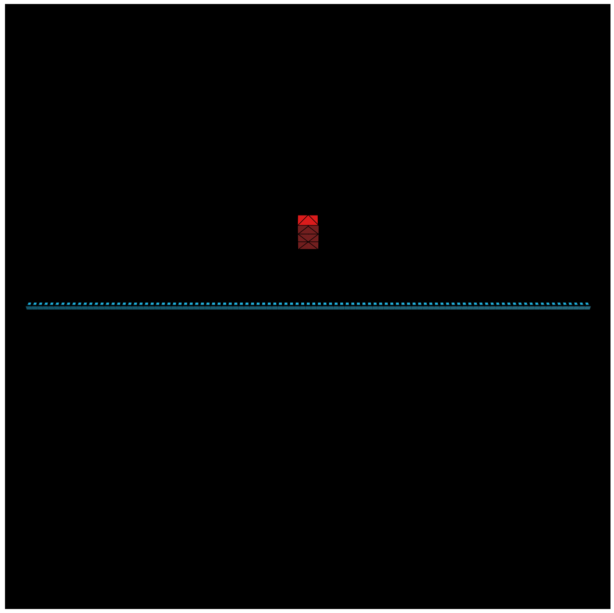


Figure 7.10: Scene with a simple loudspeaker model and a WFS array of 101 speakers.

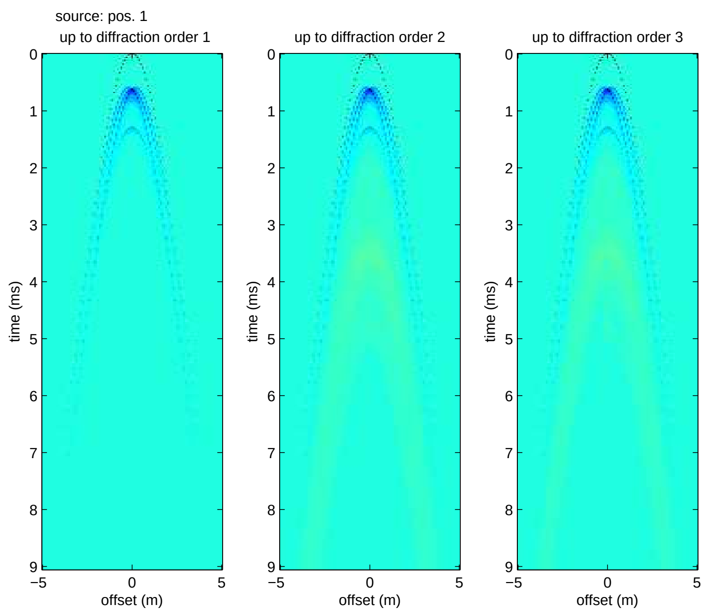
can clearly see the direct sound, followed by the first order diffractions from the edges, then followed by the second and third order diffraction. The spatial properties can clearly be distinguished.

In figure 7.11b, we can see that the direct sound is missing, which is to be expected. We can also see that the diffraction edges for the first order are not visible for the middle speakers (the ones in between -0.2 and 0.2m.).

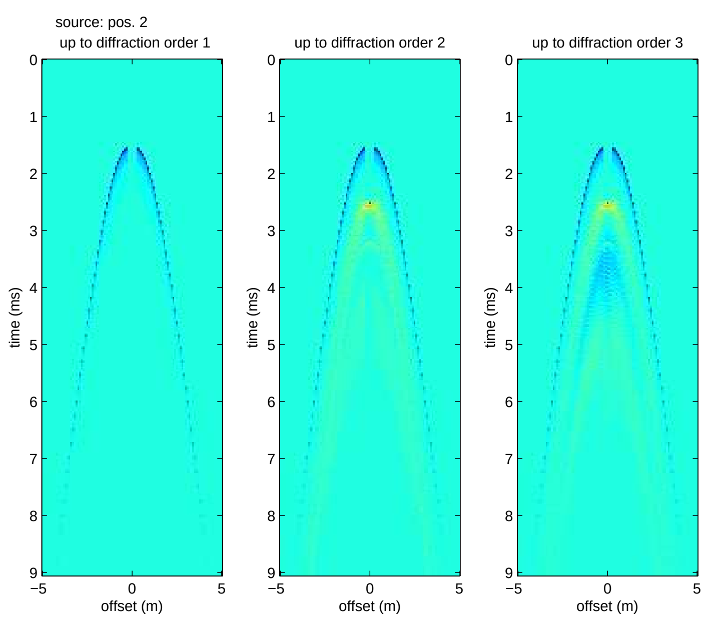
The responses of two single speakers are shown in figures 7.12 and 7.13, for a speaker on-axis ($(0, 3, 0)$m.), and a speaker off-axis ($(-2, 3, 0)$m.), for the model turned towards the reproduction array and turned away from it.

The difference between the speakers on and off-axis are clear: for the speaker on the axis, the contributions from the side and top edges coincide, whereas for the speaker off-axis, they arrive at different times. In both cases we can see a lowpass filtering effect of the diffracted signal, as well as a comb filter effect for the higher frequencies. This is the expected behaviour of the diffraction impulse response (see also [Van91, FBA03]). We see that with higher order the lowpass filtering effect gets stronger. What is especially interesting is that for frequencies above 1kHz there is only a very small difference between the different orders of calculation; the main differences are found in the low frequency region. There they vary most on off-axis locations. This would suggest that it may not be necessary to calculate the signals up to a very high order.

For the object with the source point away from the reproduction array, we can see that the diffracted sound does not arrive for the first order at the speaker



(a) Source turned towards the array



(b) Source turned away from the array

Figure 7.11: Speaker responses for the scene from figure 7.10 in the time domain for the object location $(0, 5, 0)$.

on-axis. For both on-axis and off-axis speakers the difference between the second and third order is larger than with the source point towards the reproduction array; the absence of direct (and first order diffracted) sound makes the higher orders more important. From 1kHz onwards the difference is relatively small.

7.2.2 Extrapolation into the listening area

The loudspeaker signals were convoluted with the Gaussian wavelet shown in figure 7.14, and extrapolated into the listening area, to 3m and 8m in front of the reproduction array. The resulting responses are shown in figures 7.15 to 7.18, for the two orientations of the object, and two (virtual) distances behind the reproduction array.

We can see that the spatial nature of the diffraction is kept. We can also see that the first order diffraction, which was not present for the middle speakers, is smeared out into the on-axis listening positions getting further from the reproduction array. This is logical, as the waves propagate in all directions from the speakers. In a way, it is a contradiction in the model, as the speakers will still act as a kind of "window" for the sound coming from the object, where the sound diffracts again from this window. When we compare the sound field from the object placed at 2m. behind the array at a listening distance of 8m. from the reproduction array (figure 7.16, lower plots), with the sound field from the object placed at 7m. behind the array at a listening distance of 3m. from the reproduction array (figure 7.18, upper plots), we can see that there is a clear difference between the two cases, although for the listener both situations represent a source object at 10m. distance. Whereas in the first case the response in the middle of the listening area is already forming a coherent wave front, the response in the second case still shows clear disturbances from the shadow zone speakers.

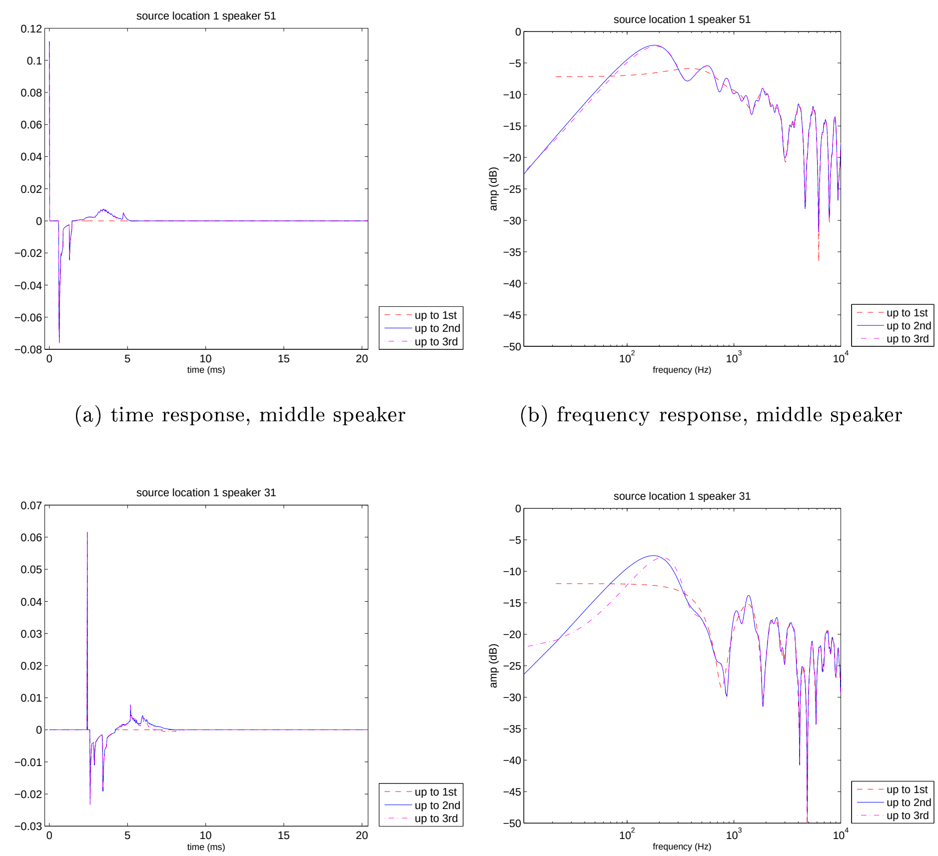
For the same comparison for the object with the source turned towards the reproduction array (figure 7.15, lower plots, and figure 7.17, upper plots), the difference is not as serious.

We can also notice in 7.16 and 7.18 that the results for the third order seem not completely symmetrical, as the left side seems stronger pronounced than the right side. It is suspected that this is due to a software bug.

7.2.3 Summary of results

The diffraction calculations give plausible results. From the example shown, it can be concluded that for source points facing the reproduction array, the diffraction only needs to be calculated up to first or second order to get an accurate result. For sources on the backside (i.e. not facing the reproduction array), the response needs to be calculated up to second or third order to get more accurate results for the low frequencies. For a more complex shaped object, higher order calculations may be needed to account for all diffraction edges.

As the diffraction model is a geometrical model, there occurs a kind of second "stage of diffraction" at the reproduction array: as the diffracted sound for some speakers may be blocked by the object for certain diffraction orders, there will



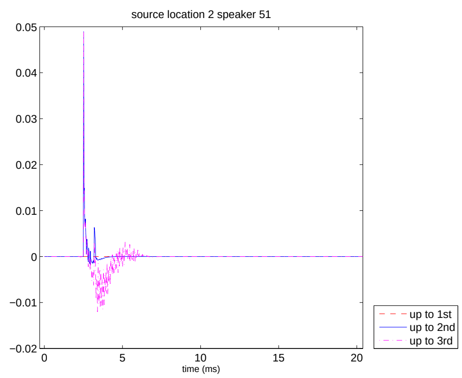
(a) time response, middle speaker

(b) frequency response, middle speaker

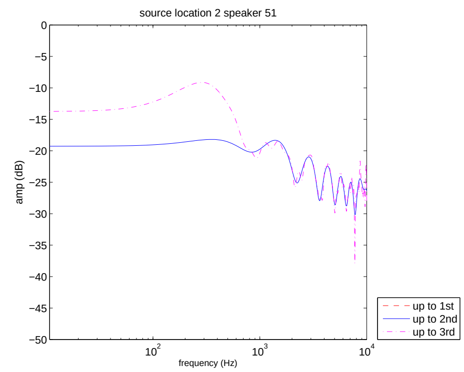
(c) time response, off-axis speaker

(d) frequency response, off-axis speaker

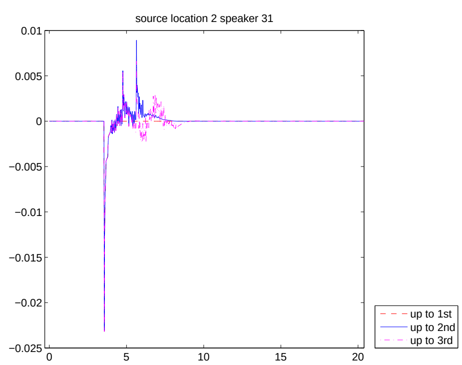
Figure 7.12: The impulse responses for the speaker at coordinates $(0, 3, 0)$ (top) and the speaker at coordinates $(-2, 3, 0)$ (bottom), for the loudspeaker model object placed at $(0, 5, 0)$ with the source turned towards the reproduction array.



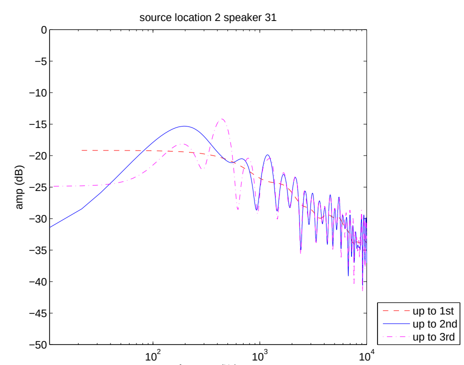
(a) time response, middle speaker



(b) frequency response, middle speaker



(c) time response, off-axis speaker



(d) frequency response, off-axis speaker

Figure 7.13: The impulse responses for the speaker at coordinates $(0, 3, 0)$ (top) and the speaker at coordinates $(-2, 3, 0)$ (bottom), for the loudspeaker model object placed at $(0, 5, 0)$ with the source turned away from the reproduction array.

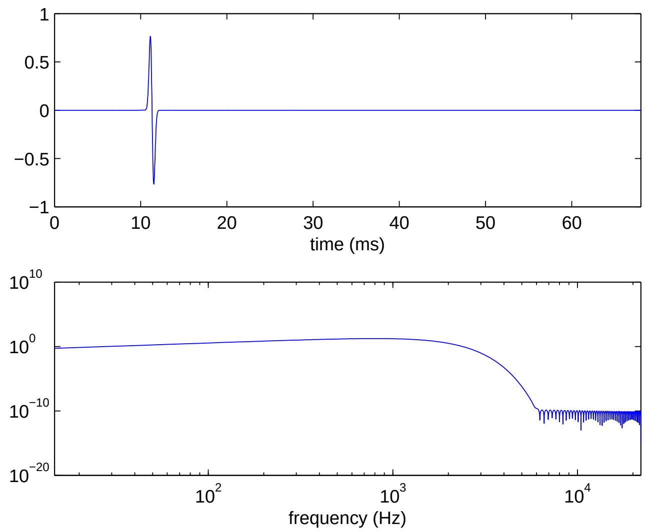


Figure 7.14: The Gaussian wavelet with which the speaker signals were convoluted before extrapolation into the listening area.

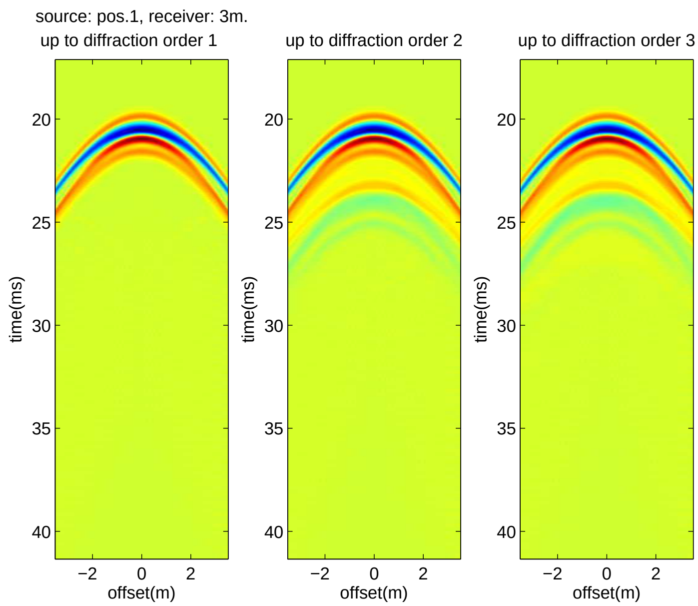
be no contribution of that order for those speakers; however upon emitting the sound into the reproduction area, the sound emitted by the speakers next to the speakers in the shadow zone, will emit their contribution also into this zone. As a consequence there will be a difference between the reproduction of a signal from an object at a (virtual) distance close to the array, listened to far away from the reproduction array, and a signal from an object positioned at a large virtual distance, listened to at a close distance from the reproduction array.

It is subject for future research in how far this is a problem, when we have not only sound coming from one point on the object, but from points all over the object. In that case the diffracted signals will play a less important role, and the effect may not be disturbing.

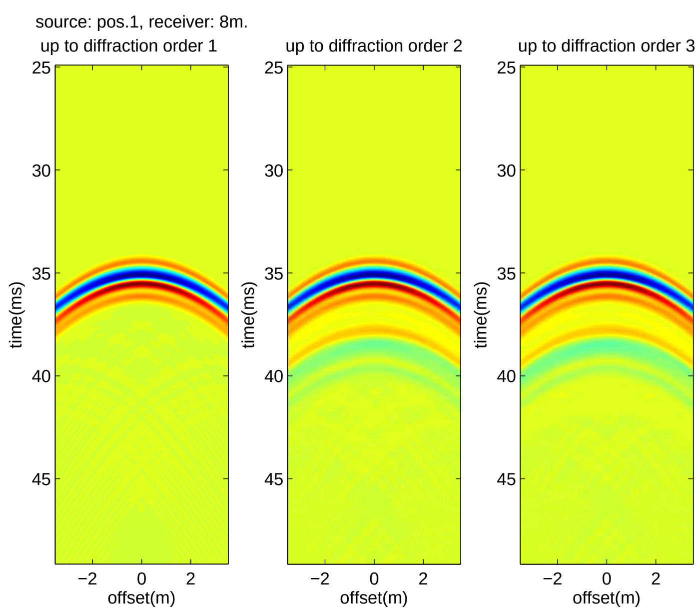
For a physically correct reproduction it may be worthwhile to formulate a different model for diffraction, which does not suffer from this geometrical error; for example a model which calculates the diffraction for a representative area in the listening area and does a backwards transformation to the loudspeaker array to calculate the reproduction signals.

7.3 Perceptual effects

To study the perceptual effects of this approach, the signals described in section 7.1 were presented in a listening test, based on the repertory grid method [BR99]. This method consists of two phases: in the first phase the test subjects are presented several signals, which they have to compare with each other and

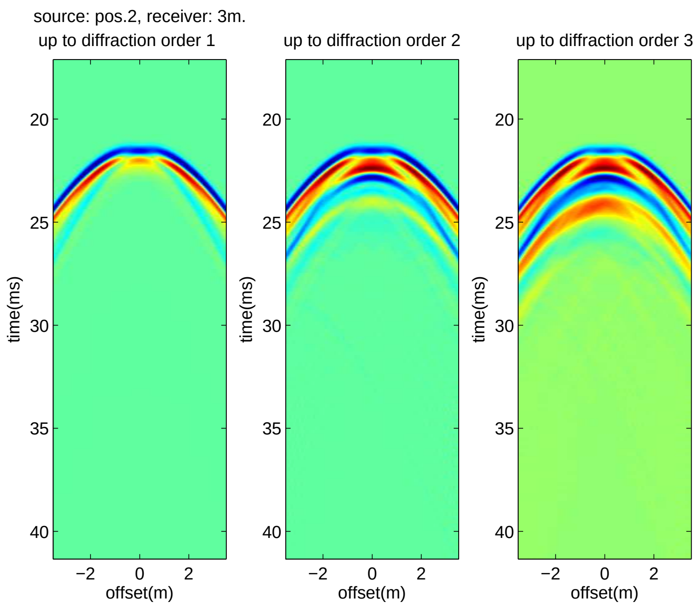


(a) At 3m distance

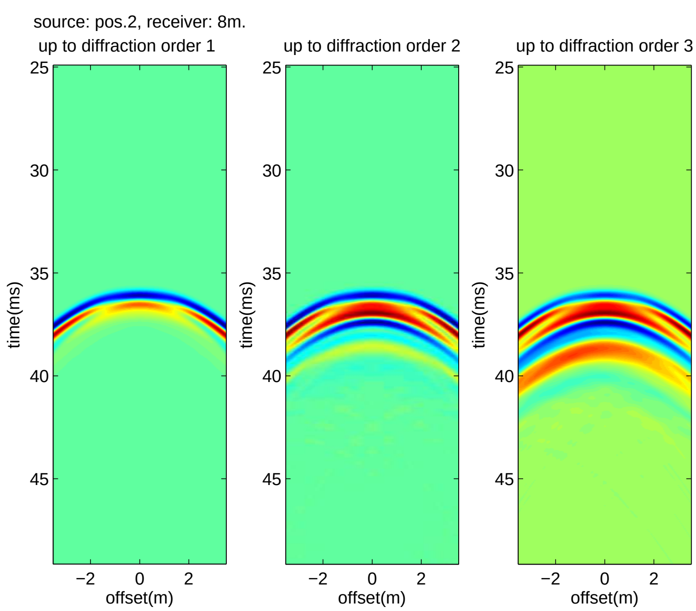


(b) At 8m distance

Figure 7.15: The extrapolated signal at 3m (top) and 8m (bottom) for the loudspeaker model at the position $(0, 5, 0)$, turned towards the listening area.

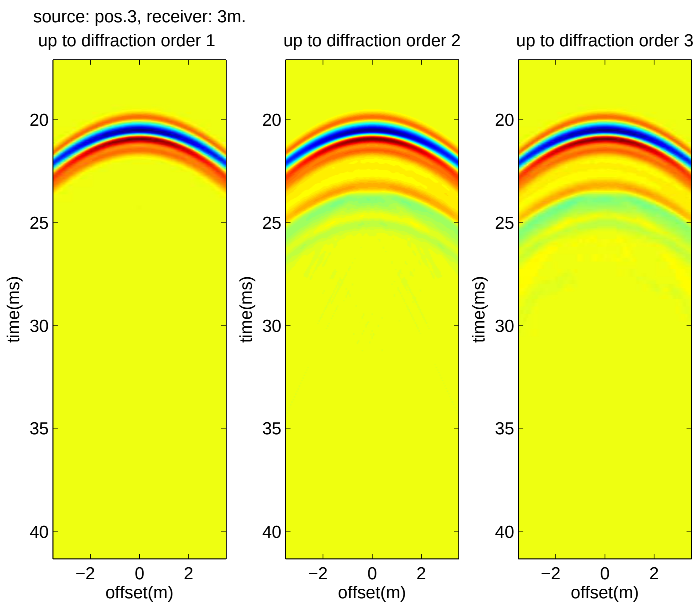


(a) At 3m distance

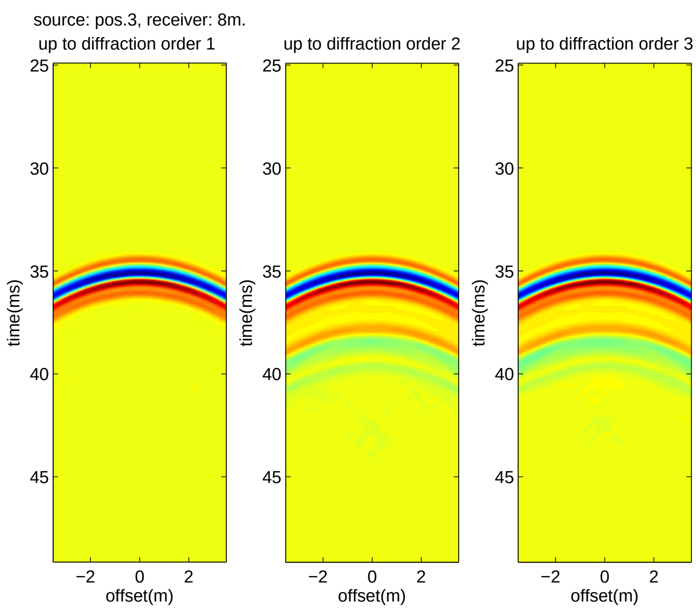


(b) At 8m distance

Figure 7.16: The extrapolated signal at 3m (top) and 8m (bottom) for the loudspeaker model at the position $(0, 5, 0)$, turned away from the listening area.

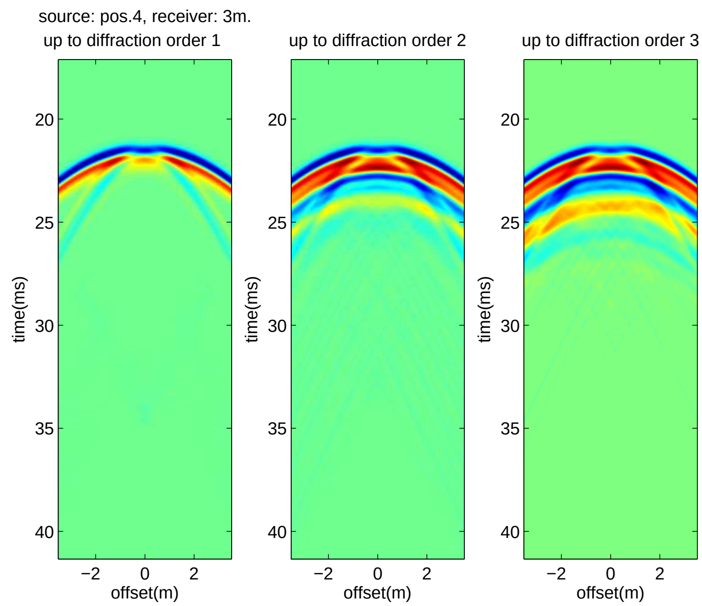


(a) At 3m distance

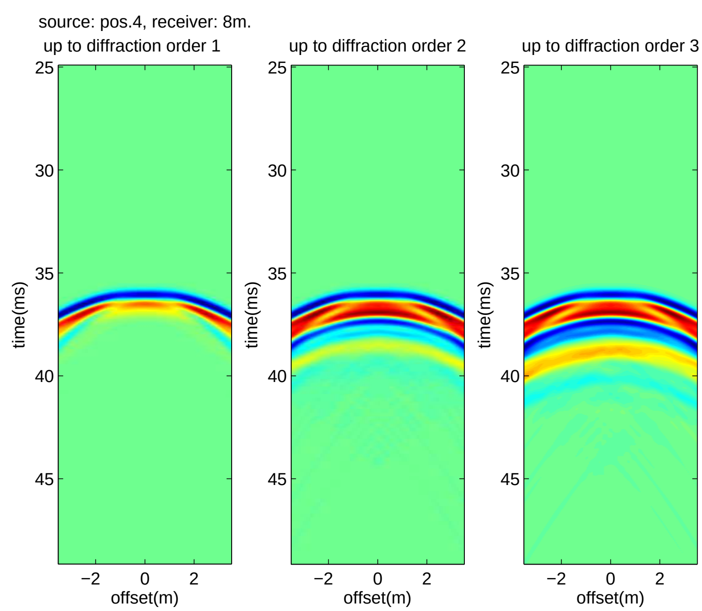


(b) At 8m distance

Figure 7.17: The extrapolated signal at 3m (top) and 8m (bottom) for the loudspeaker model at the position $(0, 10, 0)$, turned towards the listening area.



(a) At 3m distance



(b) At 8m distance

Figure 7.18: The extrapolated signal at 3m (top) and 8m (bottom) for the loudspeaker model at the position $(0, 10, 0)$, turned away from the listening area.

describe the differences between them. Often this is done in triads: of three signals, the test subject has to choose which one is the most different from the other two and describe the difference, as well as the similarity between the remaining two signals. In this way, bipolar pairs of characteristics are constructed. In the second phase of the test, each signal is quantitatively rated on those bipolar scales.

The choice for the repertory grid method was motivated by the difficulty to create reference signals, which made it impossible to compare the reproduced signals with real signals as emitted by 3-dimensional objects. Furthermore, as one of the main goals of the method is to create spatial, perceptual effects for composition of electro-acoustic music, it was found interesting to investigate the perceptual effects resulting from certain parameters in the reproduction. As the repertory grid method allows the test subjects to freely describe the perceptual effects in the first phase of the method, while, in the second phase, rating signals on bipolar scales resulting from the first phase, the results of these listening tests could give some guidelines of which effects can be achieved by changing certain parameters in the reproduction.

7.3.1 Test setup

The first series of listening tests were performed in the WFS listening room[1] of the Fraunhofer Institute in Ilmenau. The loudspeaker setup is shown in figure 7.19, and a photograph of the setup is shown in figure 7.20.

The 20 test subjects were employees of Fraunhofer and/or students at the Technical University of Ilmenau. Of them, 16 had experience with listening to WFS reproduction (and 2 a little), 8 were actively making music, 4 were female, ages varied between 18 and 43 years.

Three different kinds of material were used: a male voice, a slow acoustic guitar and a fusion drum kit. Each signal was 10 seconds long; the test subject could play the signals as often as they liked, and also stop the signal before the 10 seconds were over.

Instructions to the test subjects were given in written form (see appendix D), with a further verbal elaboration if needed by the test subject. The listening test software was created with *SuperCollider* [McC] and *swonder3Dq*, using the graphic toolbox *SwingOSC* [Rut, v. 0.50] for the user interface. The filters for the loudspeakers were first created with *swonder3Dq*. Then on two rendering machines (each driving 16 loudspeaker channels), the *swonder3Dq engine* was running (using *BruteFIR* [Tor05] for the convolution). The signals were played from a third computer running *SuperCollider*, which also sent an OSC-message to the renderers to change the position of the object to the desired location. A schematic overview of this setup is shown in figure 7.21 and a screenshot of the test interface for the first phase is shown in figure 7.22.

At the moment of the test, the amplitude factor as described in section 6.5 was not implemented yet, so the test signals were adjusted for loudness, by listening to and comparing each signal to the point source signal.

[1] Dimensions: width: 4.35 m, length: 7.75 m., height: 3.4 m., reverberation time: 0.25 s.

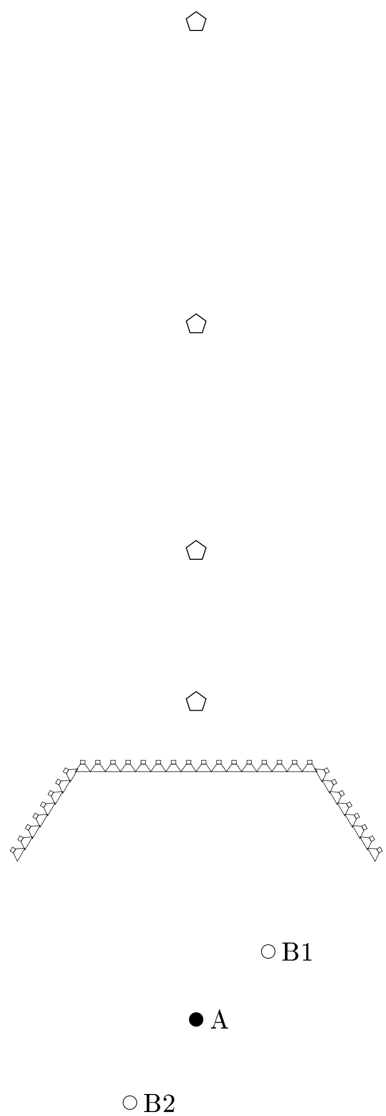


Figure 7.19: Listening test setup in the listening room at the Fraunhofer Institute utilising 32 speakers. The dot indicates the listening position for the first phase of the test; the circles indicate the listening positions for the second phase of the test. The pentagons indicate the virtual sound source position (1 m., 3 m., 6 m. and 10 m. behind the array)

Figure 7.20: Listening test setup in the WFS listening room of the Fraunhofer institute. Here the view from listening position 2 of the second phase of the listening test is shown.

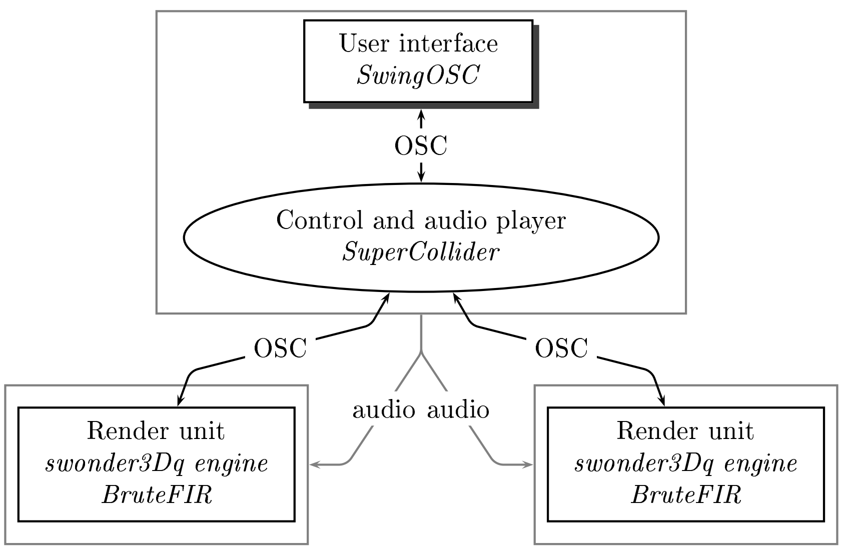


Figure 7.21: Schematic overview of the software and computer hardware used in the listening tests.

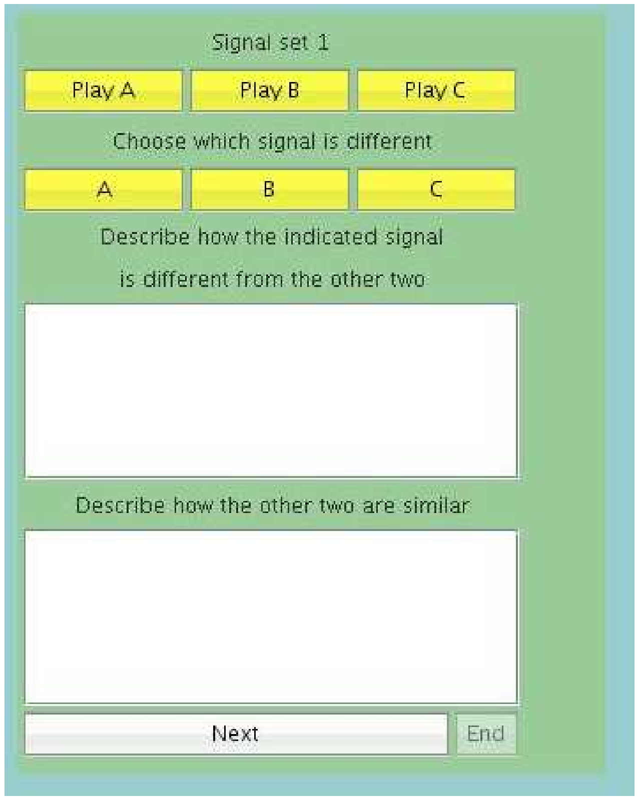


Figure 7.22: Test subject interface for the first phase of the repertory grid method. *Note:* as the test was performed in Germany, the labels were translated into German for the test.

Though in this experiment the shapes used are the same as in the previous section, the sizes of the object differed: the sphere had a diameter of 2m., the tetrahedron of 3.2m., and the icosahedron of 3.4m.

7.3.2 Results of the first part

The results of the first phase were analysed by reading the descriptions given, and extracting the most common ones for all test subjects.

The following descriptive pairs[2] were extracted and used in the second phase of the experiment:

- dull (dumpf) - clear (klar),
- dark (dunkel) - bright (hell),
- thin (dünn) - full (voll),

[2]The terms were given by the test subjects in German; here a translation into English by the author is given, as well as the actual German descriptions in brackets.

- no bass (kein Bass) - a lot of bass (viel Bass),
- no high tones/treble (keine Höhen) - a lot of high tones/treble (viel Höhen),
- quiet (leise) - loud (laut),
- narrow (eng) - broad (breit),
- left (links) - right (rechts),
- near (nah) - far (weit),
- below (unten) - above (oben),
- dry (trocken) - spacious (räumlich),
- clear location (klare Ortung) - diffuse location (diffus).

The first five descriptors listed here are related to the tone colour (timbre), the last six to spatial dimensions. It is interesting to see that some of the test subjects distinguished differences in the vertical direction as well as in the horizontal direction.

Another descriptor that was often used was naturalness. Other things some participants noted were comb filter effects for some signals and differences in localisation for different instruments within the sound sample.

The choice in different materials was good, as they emphasised different perceptual characteristics.

7.3.3 Semantic differential (phase 2)

The descriptive pairs listed in the previous section were used in the second phase of the listening test in order to analyse how the different parameters of the object definition affect the perception of these objects. The pairs were the same for each test subject.

For this part the following sets were used:

Shape point source, icosa (2 times refined), tetrahedron (2 times refined), sphere.

Size icosa, icosa half, icosa quart, icosa (3 times refined), icosa half (2 times refined).

Refinement icosa, icosa (2 times refined), icosa (3 times refined), icosa (4 times refined).

Each set was presented at two distances, and with two stimuli (the voice and the guitar from the first part), and the test subjects had to listen to all the sets twice: once from position 1 (right-front) and once from position 2 (left-back).

The order of the sets was randomised, as was the order of the signals within the set. The test subject could switch to another signal while it was playing using a crossfade, and could listen to the signals as often as they wanted. The test signals were 20 seconds long. The test interface is shown in figure 7.23.

Instructions to the test subjects were given in written form (see appendix D), with a further verbal elaboration if needed by the test subject. Apart from the test interface, the software used was the same as in the first part of the experiment.

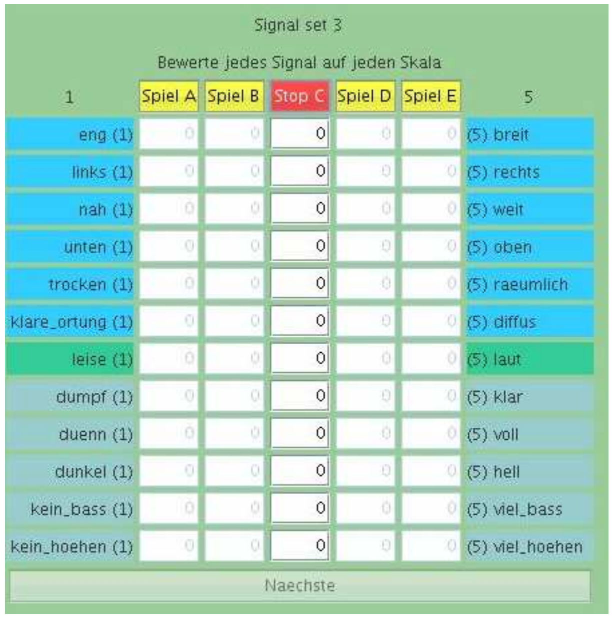


Figure 7.23: Test subject interface for the semantic differential.

19 test subjects participated (again employees of Fraunhofer and/or students at the Technical University of Ilmenau), of whom 13 did the listening test at both positions, 3 only at position 1 and 3 only at position 2. All but one listened to all sets; the one who did not, listened to one set twice, and did not listen to another set (due to a technical error that occurred during the test). 15 of the test subjects also participated in the first part of the test, so they were somewhat familiar with the signals. 14 had experience with listening to WFS reproduction (and 3 a little), 7 were actively making music, 4 were female, ages varied between 24 and 41 years.

7.3.4 Analysis of the results

The data was analysed with *R* [R D07], after it had been converted with *Octave* [Eat02] to a usable format. These sets were analysed separately: shape, size and refinement was regarded separately; the listening position, the distance and the stimulus type were regarded as factors in the analysis.

7.3.4.1 Refinement

In table 7.1 a summary is given of which factors have significant influence on the perception of each feature, for the refinement set, based on the ANOVA tables which can be found in appendix E.

Based on these dependencies, the means for each feature are displayed in figures 7.24 and 7.25.

feature	factors
eng - breit	signal, position
links - rechts	signal, position, dist:sig:pos, distance, dist:sig, pos:sig
nah - weit	dist:sig, dist:stim, position, signal
unten - oben	position, signal
trocken - räumlich	signal, stimulus, position
klare Ortung - diffus	signal, distance
dumpf - klar	signal, sig:stim, pos:sig, position, stimulus
dünn - voll	signal
dunkel - hell	signal, dist:stim
kein - viel Bass	signal, position
kein - viel Höhen	signal, distance
leise - laut	signal, dist:sig, distance, pos:sig

Table 7.1: Overview of the features, and which factors have significant influence on them, in order of strength, for the refinement set. Interaction factors are separated by ":". Abbreviations used are: dist(ance), pos(ition), sig(nal), stim(ulus).

For the feature *eng-breit*, we can see that the listeners could distinguish between the non-refined object, and the refined versions, but not between different refined versions. At the listening position more to the back, the objects are perceived as slightly less narrow, which is to be expected.

For the feature *links-rechts*, it can be observed that there is a slight difference in perception, depending on the listening position. For higher refinement, the position of the object is observed to be more central. Also for a larger distance (the second plot in figure 7.24b) of the source, there is almost no variation in the perception of the horizontal placement of the source.

For the feature *nah-weit*, there is a slight difference in perception for the unrefined object, depending on the listening position; at a close distance of the object, it is observed to be further away with higher refinement, whereas at a high (virtual) distance of the object, the trend is almost reversed.

The feature *unten-oben* does not vary significantly between the signals.

The feature *trocken-räumlich* is distinguished between non-refined and refined, and varies depending on the stimulus and the listening position.

With increasing refinement the objects are perceived to be more diffuse. This effect is slightly less when the object is at a higher distance.

The loudness of the signals does not vary significantly (except for the unrefined object) between the distances. At the listening position farthest away almost no difference is perceived, whereas at the close listening position the more refined objects are perceived to be louder. For the objects at a virtually close distance the loudness is more equal, whereas for a large virtual distance, the object seems to get a bit louder with increasing refinement.

The unrefined object was perceived to be significantly less "full" (*dünn-voll*),

correlation	ortung	klar	voll	hell	bass	hoehen
klare ortung diffus	1.000	**−0.576**	0.347	−0.492	**0.526**	−0.421
dumpf klar	**−0.576**	1.000	**−0.503**	**0.852**	**−0.780**	**0.700**
duenn voll	0.347	**−0.503**	1.000	**−0.530**	**0.676**	−0.396
dunkel hell	−0.492	**0.852**	**−0.530**	1.000	**−0.798**	**0.732**
kein bass viel bass	**0.526**	**−0.780**	**0.676**	**−0.798**	1.000	**−0.682**
kein hoehen viel hoehen	−0.421	**0.700**	−0.396	**0.732**	**−0.682**	1.000

Table 7.2: Correlation between different features (the full table can be found in appendix E) for the refinement set.

than the refined versions, where there is an increase in fullness perceived with higher refinement, but no difference between the last two refinements.

The brightness (*dumpf-klar*) decreases clearly with increasing refinements, with slight variations depending on the listening position and on the stimulus. Similar effects are seen for the feature *dunkel-hell*, and the amount of treble/highs. The amount of perceived bass increases with increasing refinement.

From this discussion and the figures, we can see that there seems to be a high correlation between the various tone colour features; this is confirmed, when we calculate the correlation between the factors, as shown in table 7.2.

To further investigate how many features are actually relevant, a principal component analysis was done [Bor05]. The results are shown in table 7.3. From this we see that according to the Kaiser-Guttmann criterion[3] 3 components are relevant; this is confirmed by looking at the Scree plot in figure 7.32a. When looking at biplots for these components, we can get an idea of what the components represent, by looking at which components are correlated to which features. These biplots are shown in figure 7.33a. Here we can see that the first component consists mainly of the features bass, full, treble, brightness, clarity, whereas the second component is related to source width and spaciousness, and the third to localisation.

7.3.4.2 Shape

In table 7.4 a summary is given of which factors have significant influence on the perception of each feature, for the shape set, based on the ANOVA tables which can be found in appendix E.

Based on these dependencies, the means for each feature are displayed in figures 7.26, 7.27 and 7.28.

For the feature *eng-breit*, we can clearly see that a point source is perceived more narrow than a complex object. For the different objects (other than the point source), there is a difference in perception, based on the listening position and the distance of the object.

[3] according to which the variance has to be larger than 1.

```
Importance of components:
                          Comp.1    Comp.2     Comp.3    Comp.4
Standard deviation     2.1188500  1.3130143  1.08349409 0.9413208
Proportion of Variance 0.3741271  0.1436672  0.09782995 0.0738404
Cumulative Proportion  0.3741271  0.5177943  0.61562430 0.6894647
                          Comp.5    Comp.6     Comp.7     Comp.8
Standard deviation     0.8914157  0.83893611 0.77252678 0.74397199
Proportion of Variance 0.0662185  0.05865115 0.04973314 0.04612453
Cumulative Proportion  0.7556832  0.81433435 0.86406748 0.91019201
                         Comp.9    Comp.10     Comp.11    Comp.12
Standard deviation     0.68831589 0.54429286 0.41462863 0.36843655
Proportion of Variance 0.03948156 0.02468789 0.01432641 0.01131212
Cumulative Proportion  0.94967357 0.97436147 0.98868788 1.00000000
```

Table 7.3: Importance of components in the principal component analysis for the refinement set.

For the feature *nah-weit*, we can see that a point source is perceived to be closer, when the objects are at a larger distance.

For the left-right position, there is a clear distinction between the different objects at the front listening position, whereas with increasing distance of the object, some objects are perceived differently.

There is again almost no variation in the perception in the vertical direction (*oben-unten*).

The point source is perceived to be more dry than the complex objects, with slight differences for the different stimuli. A similar effect occurs for the *klare Ortung-diffus* feature, where there is also a slight dependency on the distance of the object.

There is a significant difference in perceived loudness between the objects, while also the stimulus and distance play a role in the perceived loudness.

For the feature *dumpf-klar*, we can see that there is a clear distinction between the point source, the sphere, and the icosa/tetrahedron. At a larger distance, the distinguishment is more clear. A similar effect occurs for the features *dunkel-hell*, *kein-viel Bass* and *kein-viel Höhen*. For the feature *dünn-voll*, the effect is also similar, except for the tetrahedron at a large distance, which is perceived to be more full.

As with the refinement set, we notice that there is a strong correlation between the tone colour features, as shown in table 7.5.

From the principal component analysis (see table 7.6), we see that according to the Kaiser-Guttmann criterion 3 components are relevant, which is confirmed by the Scree-test (figure 7.32b). From the biplots (figure 7.33b) we can see that again the first component is connected to the tone colour, whereas the other components are more related to spatial features.

feature	**factors**
eng - breit	signal, distance, pos:sig
links - rechts	signal, dist:pos, pos:sig, dis:pos:sig, dist:sig, position
nah - weit	signal, dist:sig
unten - oben	signal
trocken - räumlich	signal, stimulus
klare Ortung - diffus	signal, dist:sig, sig:stim
dumpf - klar	signal, dist:sig, dist:stim
dünn - voll	signal, dist:sig
dunkel - hell	signal, dist:sig, distance
kein - viel Bass	signal, dist:sig, distance, pos:stim
kein - viel Höhen	signal, dist:sig, distance, stimulus
leise - laut	signal, dist:sig, distance, dist:sig:stim

Table 7.4: Overview of the features, and which factors have significant influence on them, in order of strength, for the shape set. Interaction factors are separated by ":". Abbreviations used are: dist(ance), pos(ition), sig(nal), stim(ulus).

correlation	ortung	klar	voll	hell	bass	hoehen
klare ortung diffus	1.000	−0.324	0.351	−0.287	0.368	−0.258
dumpf klar	−0.324	1.000	−0.306	**0.689**	**−0.523**	0.477
duenn voll	0.351	−0.306	1.000	−0.442	**0.729**	−0.361
dunkel hell	−0.287	**0.689**	−0.442	1.000	**−0.615**	**0.588**
kein-viel bass	0.368	**−0.523**	**0.729**	**−0.615**	1.000	**−0.503**
kein-viel hoehen	−0.258	0.477	−0.361	**0.588**	**−0.503**	1.000

Table 7.5: Correlation between different features (the full table can be found in appendix E) for the shape set.

7.3.4.3 Size

In table 7.7 a summary is given of which factors have significant influence on the perception of each feature, for the size set, based on the ANOVA tables which can be found in appendix E.

Based on these dependencies, the means for each feature are displayed in figures 7.29, 7.30 and 7.31.

The perceived width of the source is not clearly dependent on the size of the object. The full size, finely refined object is clearly perceived to be wide, but the non-refined full and half size objects are perceived to be more narrow than the quarter size and half size refined versions. The differences are more clearly pronounced for the voice stimulus.

For the horizontal localisation we can see there is a clear agreement for the full size object versus the smaller objects. The localisation is also clearly

Importance of components:

	Comp.1	Comp.2	Comp.3	Comp.4
Standard deviation	1.9445795	1.2619726	1.1175722	0.97056103
Proportion of Variance	0.3151158	0.1327146	0.1040806	0.07849906
Cumulative Proportion	0.3151158	0.4478303	0.5519110	0.63041004

	Comp.5	Comp.6	Comp.7	Comp.8
Standard deviation	0.95516040	0.88828233	0.83597022	0.7902018
Proportion of Variance	0.07602762	0.06575379	0.05823718	0.0520349
Cumulative Proportion	0.70643765	0.77219144	0.83042863	0.8824635

	Comp.9	Comp.10	Comp.11	Comp.12
Standard deviation	0.71862817	0.63858442	0.51878946	0.46591694
Proportion of Variance	0.04303554	0.03398251	0.02242854	0.01808988
Cumulative Proportion	0.92549907	0.95948157	0.98191012	1.00000000

Table 7.6: Importance of components in the principal component analysis for the shape set.

dependent on the listening position, and also somewhat on the distance of the object.

The distance perception (*nah-weit*) does not seem to be very significant between the different sizes, though there is a significant difference between the sizes, depending on the distance of the object.

The vertical localisation does again not really significantly vary with the signals, though there are slight differences depending on the factors distance and stimulus.

The trend of distinguishing between the objects based on the feature *trocken-räumlich*, seems to be more dependent on the discretisation distance, than on the size itself. There is also a significant difference between the type of stimulus, and to a lesser extent the listening position.

The discretisation distance is the most dominant in the perception of the feature *klare Ortung-diffus*. There is a clear distinguishment between the signals, but it does not increase linearly with increase of size (keeping the discretisation constant).

The loudness is varying depending on the distance. Whereas for a close distance, the signals cannot be distinguished in loudness, on a large distance they can.

The feature *dumpf-klar* seems to be mostly dependent on discretisation distance, and also varies based on stimulus and distance of the object. Similar trends are seen for the features *dünn-voll*, *dunkel-hell*, *kein-viel Bass* and *kein-viel Höhen*.

So again we see a strong correlation between the tone colour features, which is verified in the correlation table 7.8.

From the principal component analysis (see table 7.9), we see that according to the Kaiser-Guttmann criterion 3 components are relevant, which is confirmed

feature	**factors**
eng - breit	signal, stimulus, sig:stim
links - rechts	dist:pos:sig, pos:sig, signal, dist:pos, dist:sig, position, distance, pos:sig:stim
nah - weit	dist:sig, sig:stim, signal
unten - oben	stimulus, distance, signal, position
trocken - räumlich	signal, stimulus, position
klare Ortung - diffus	signal, dist:sig, stim, pos:stim, sig:stim, distance
dumpf - klar	signal, distance, dist:sig, stimulus, sig:stim
dünn - voll	signal, dist:sig, distance, position, stimulus
dunkel - hell	signal, distance, dist:sig, dist:pos:sig
kein - viel Bass	signal, distance, dist:sig, sig:stim, stimulus, dist:stim
kein - viel Höhen	signal, dist:sig, distance
leise - laut	signal, dist:sig

Table 7.7: Overview of the features, and which factors have significant influence on them, in order of strength, for the size set. Interaction factors are separated by ":". Abbreviations used are: dist(ance), pos(ition), sig(nal), stim(ulus).

correlation	ortung	klar	voll	hell	bass	hoehen
klare ortung diffus	1.000	**−0.529**	0.405	−0.446	0.484	−0.333
dumpf klar	**−0.529**	1.000	**−0.512**	**0.748**	**−0.681**	**0.573**
duenn voll	0.405	**−0.512**	1.000	**−0.530**	**0.731**	−0.431
dunkel hell	−0.446	**0.748**	**−0.530**	1.000	**−0.659**	**0.607**
kein-viel bass	0.484	**−0.681**	**0.731**	**−0.659**	1.000	**−0.550**
kein-viel hoehen	−0.333	**0.573**	−0.431	**0.607**	**−0.550**	1.000

Table 7.8: Correlation between different features (the full table can be found in appendix E) for the size set.

by the Scree-test (figure 7.32c). From the biplots (figure 7.33c) we can see that again the first component is connected to the tone colour, whereas the other components are more related to spatial features.

7.3.5 Summary of results

Summarising the results, we can conclude that the difference in tone colour was the most clear perceivable feature when auralising complex objects. It changes with increasing refinement; this is in accordance with what we expected from the physical effects discussed in section 7.1.

The other perceptual components are of a spatial nature.

The feature *unten-oben* was not significant in any set, so does not seem to be an important feature; which is not surprising, as the reproduction only takes

```
Importance of components:
                          Comp.1    Comp.2     Comp.3     Comp.4
Standard deviation     2.0851601  1.2577217 1.08055942 0.96487530
Proportion of Variance 0.3623244  0.1318220 0.09730072 0.07758203
Cumulative Proportion  0.3623244  0.4941464 0.59144711 0.66902914
                          Comp.5    Comp.6     Comp.7     Comp.8
Standard deviation     0.92974503 0.82728904 0.77773513 0.72755054
Proportion of Variance 0.07203549 0.05703393 0.05040599 0.04411082
Cumulative Proportion  0.74106463 0.79809856 0.84850455 0.89261537
                          Comp.9    Comp.10    Comp.11    Comp.12
Standard deviation     0.64775185 0.6384331  0.50331123 0.45619526
Proportion of Variance 0.03496521 0.0339664  0.02111018 0.01734284
Cumulative Proportion  0.92758057 0.9615470  0.98265716 1.00000000
```

Table 7.9: Importance of components in the principal component analysis for the size set.

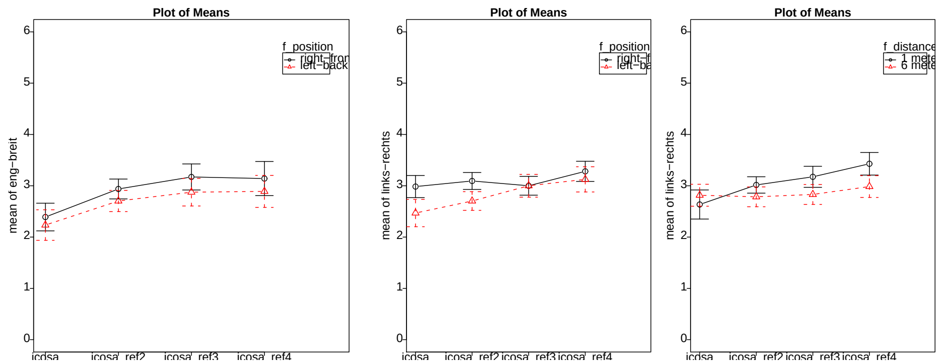
place in the horizontal plane.

With increasing refinement, the localisation was more centralised (which is in accordance with Blauert's discussion of perception of correlated sources (see also 1.3.3)), though at the same time it was considered as more diffuse, and less clearly localised.

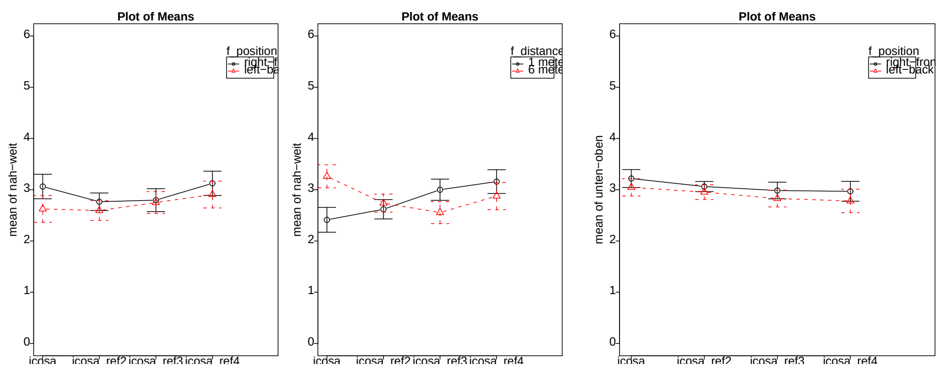
The perceived loudness differences that were found are quite likely related to the fact that although the loudness was equalised by listening to each signal, there were some signals, which were still louder. However, since the execution of the listening test, the calculation method has been expanded with an amplitude correction, which provides a better solution for the loudness equalisation.

It is suggested for future work to do another experiment, where the signals from the points on the surface are not identical, but rather dependent on each other in a physical meaningful way. The number of features tested can be limited to one or two tone colour features, and the spatial features used here, with the exception of *oben-unten*. Another feature that could be interesting to investigate is *naturalness*.

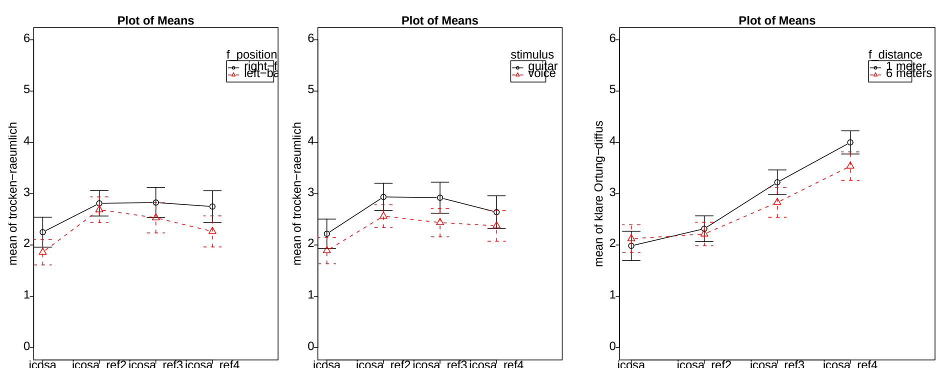
The use of recordings of voice, guitar and drums, should also be reviewed. As each of the recordings is made at a certain distance from the source itself, natural dampening of the higher frequencies will already have taken place, so the final result with this reproduction method will be exaggerated. It may be worthwhile looking into source signals which more accurately fit the source, i.e. recordings from the surface of an instrument, or source signals created through physical modelling.



(a) eng - breit (b) links - rechts

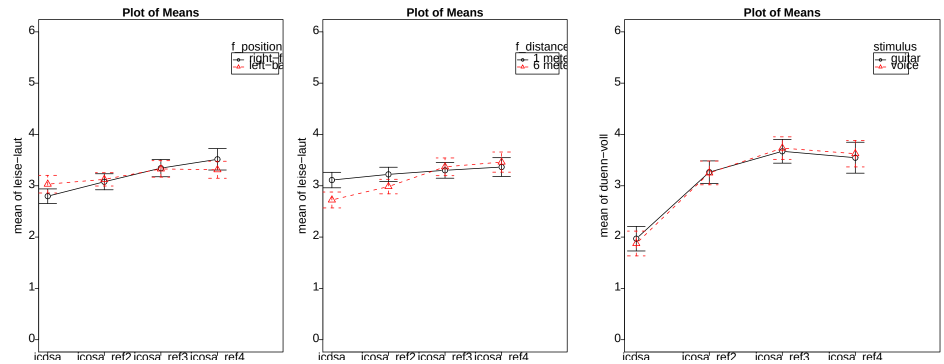


(c) nah - weit (d) unten - oben

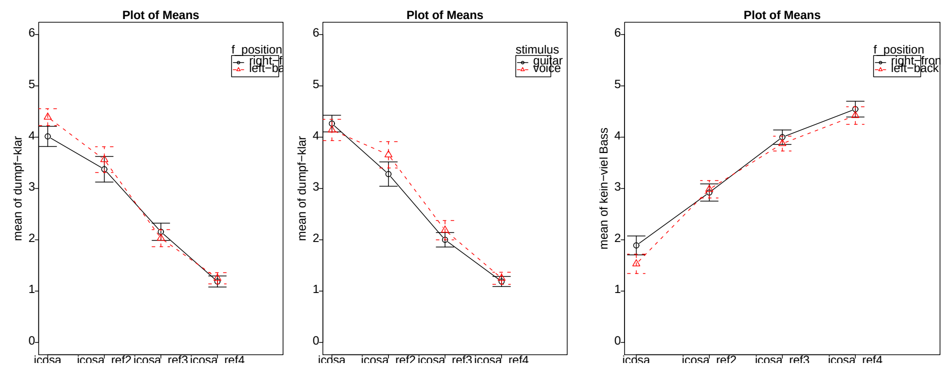


(e) trocken - räumlich (f) klare Ortung - diffus

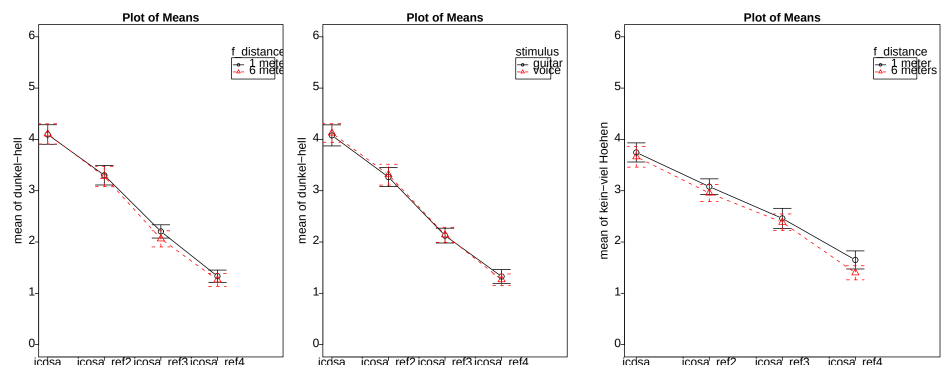
Figure 7.24: Means and 95% confidence intervals for each spatial feature, split up according to factors they are dependent on, for the refinement set.



(a) leise - laut (b) dünn - voll

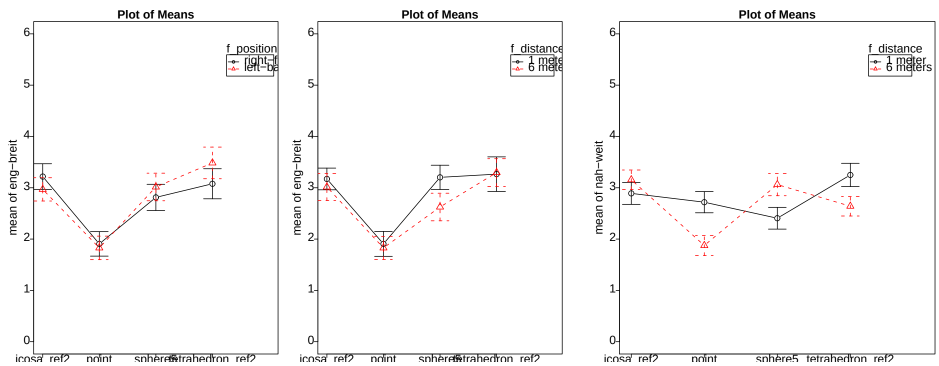


(c) dumpf - klar (d) kein - viel Bass

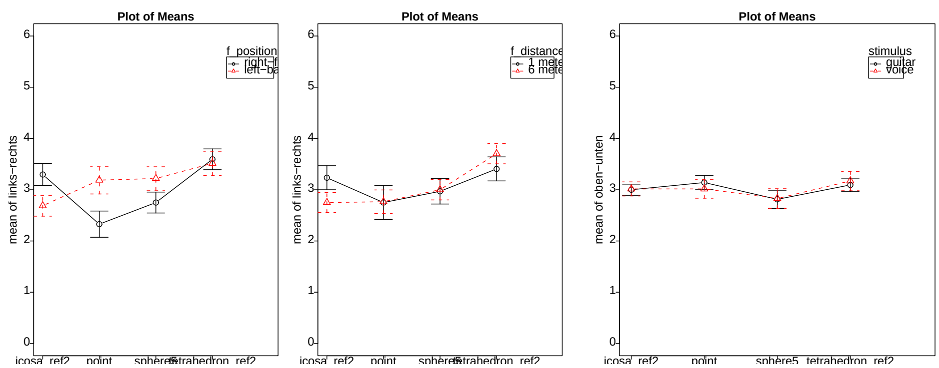


(e) dunkel - hell (f) kein - viel Höhen

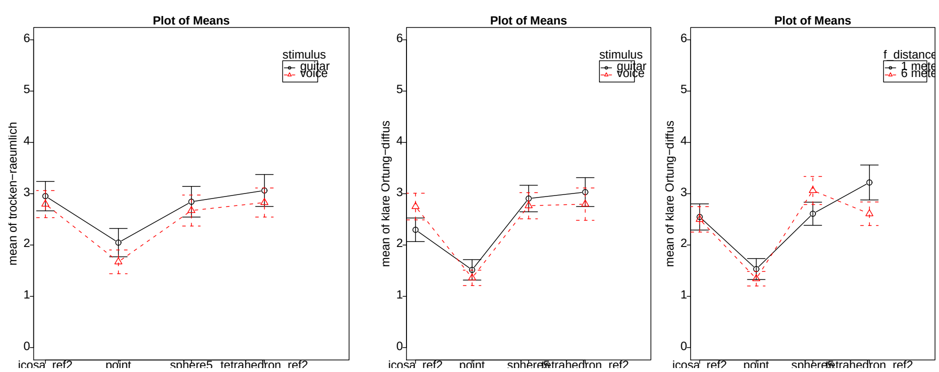
Figure 7.25: Means and 95% confidence intervals for each feature, split up according to factors they are dependent on, for the refinement set.



(a) eng - breit (b) nah - weit

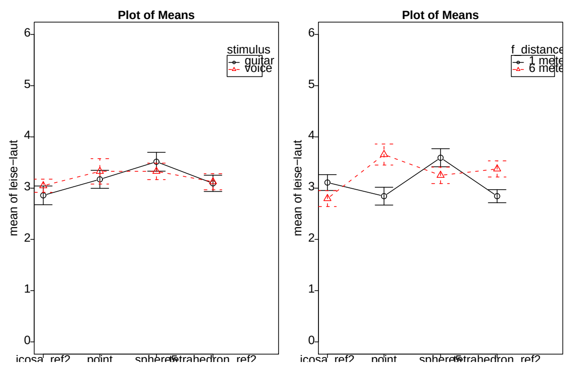


(c) links - rechts (d) unten - oben



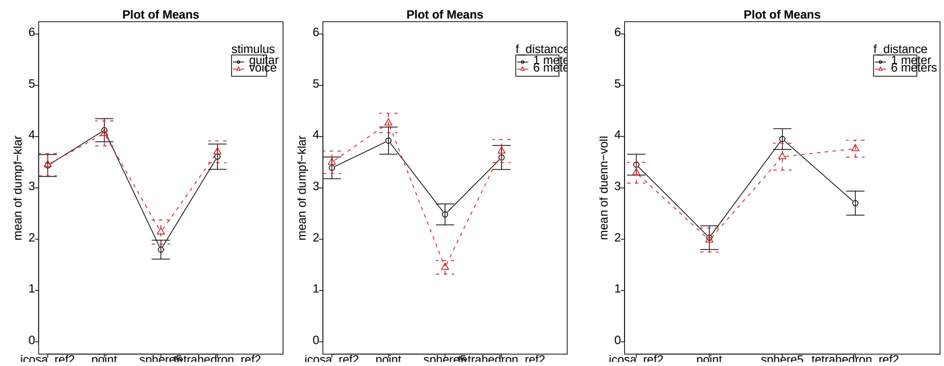
(e) trocken - räumlich (f) klare Ortung - diffus

Figure 7.26: Means and 95% confidence intervals for each spatial feature, split up according to factors they are dependent on, for the shape set.

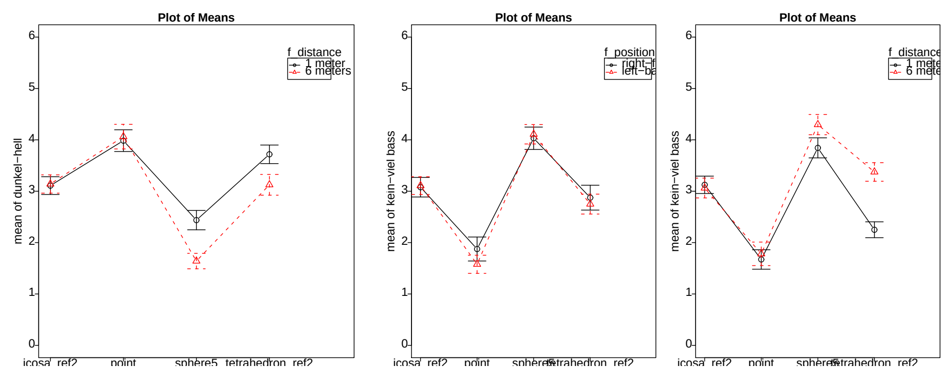


(a) leise - laut

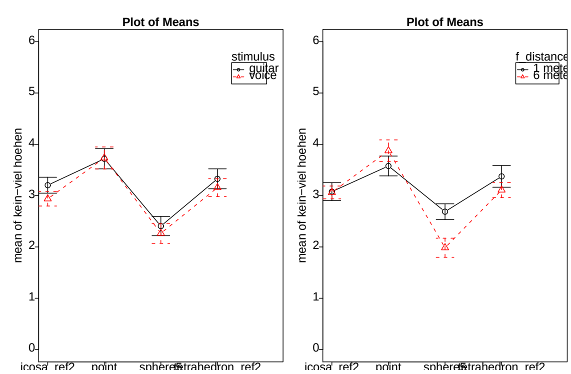
Figure 7.27: Means and 95% confidence intervals for the feature *leise-laut*, split up according to factors it is dependent on, for the shape set.



(a) dumpf - klar (b) dünn - voll

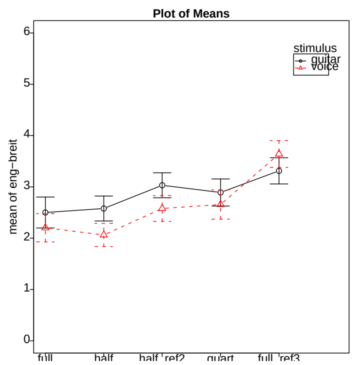


(c) dunkel - hell (d) kein - viel Bass

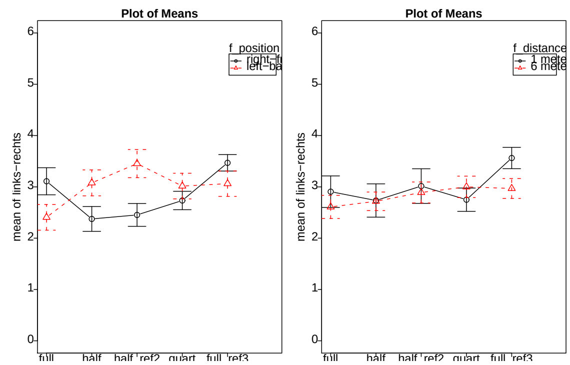


(e) kein - viel Höhen

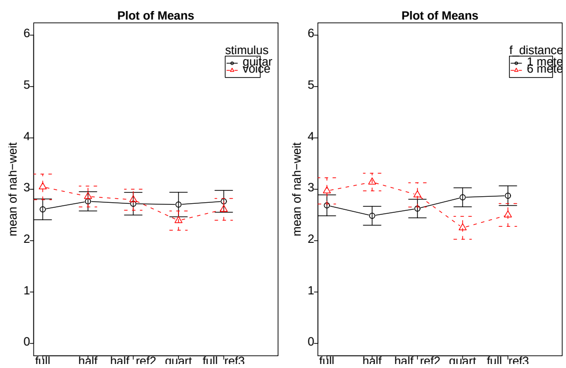
Figure 7.28: Means and 95% confidence intervals for each feature, split up according to factors they are dependent on, for the shape set.



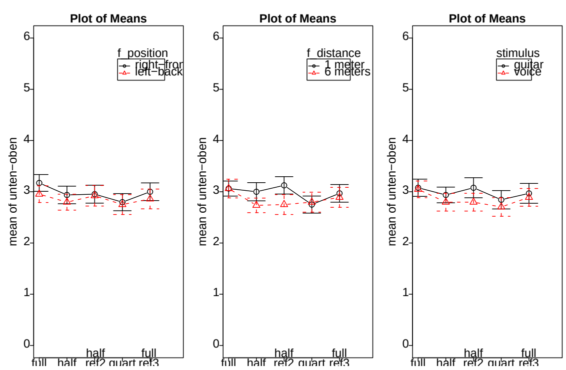
(a) eng - breit



(b) links - rechts

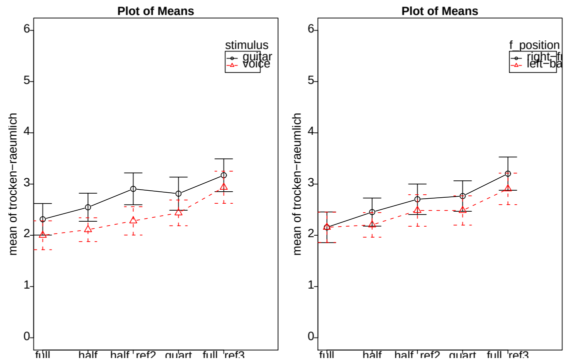


(c) nah - weit

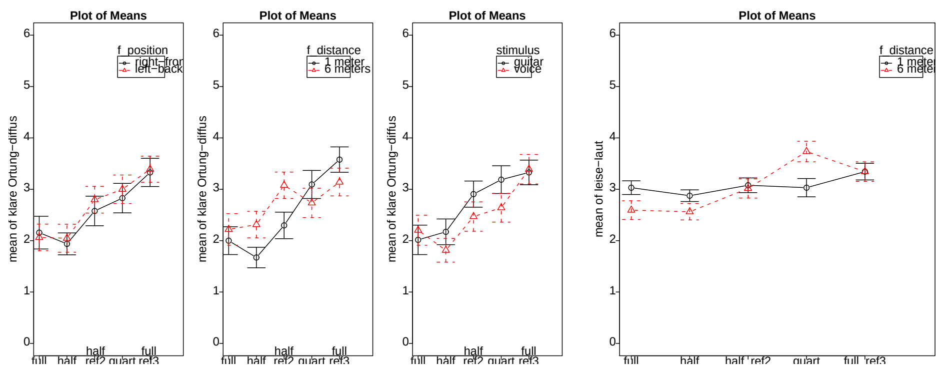


(d) unten - oben

Figure 7.29: Means and 95% confidence intervals for each spatial feature, split up according to factors they are dependent on, for the size set.

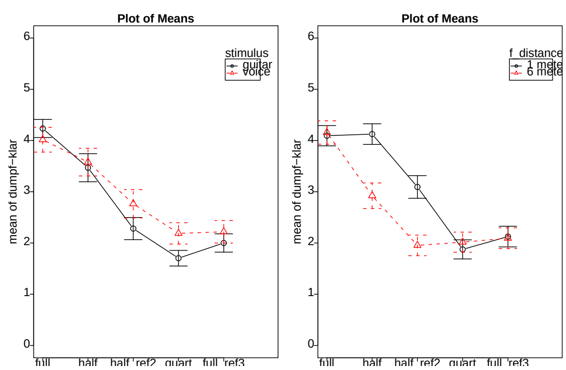


(a) trocken - räumlich



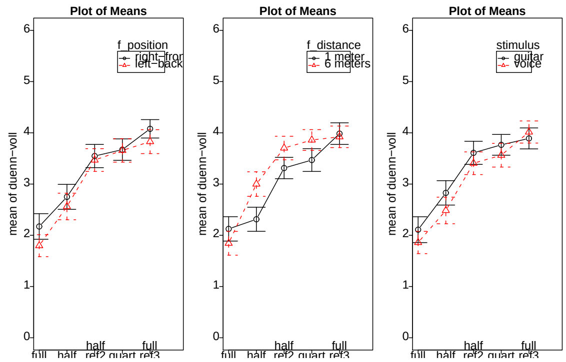
(b) klare Ortung - diffus

(c) leise - laut

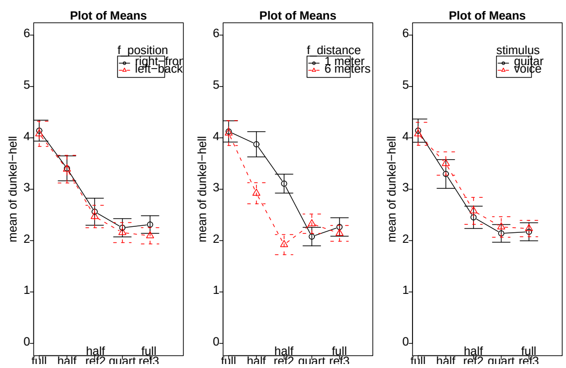


(d) dumpf - klar

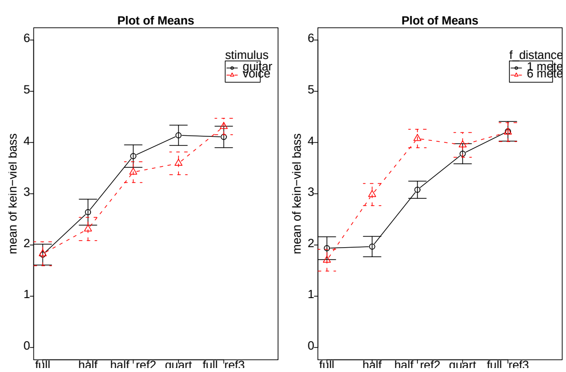
Figure 7.30: Means and 95% confidence intervals for each feature, split up according to factors they are dependent on, for the size set.



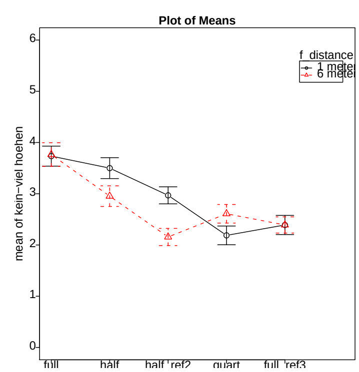
(a) dünn - voll



(b) dunkel - hell

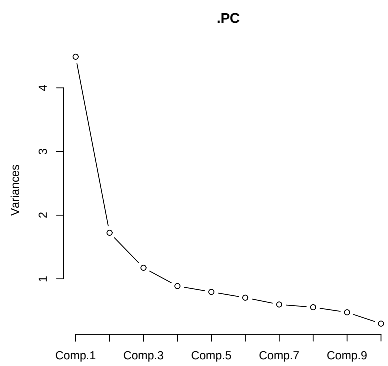


(c) kein - viel Bass

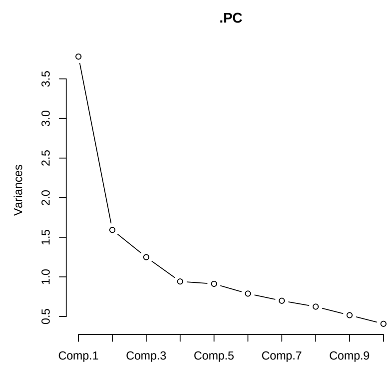


(d) kein - viel Höhen

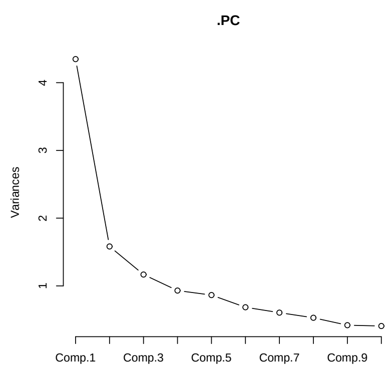
Figure 7.31: Means and 95% confidence intervals for each feature, split up according to factors they are dependent on, for the size set.



(a) Refinement set

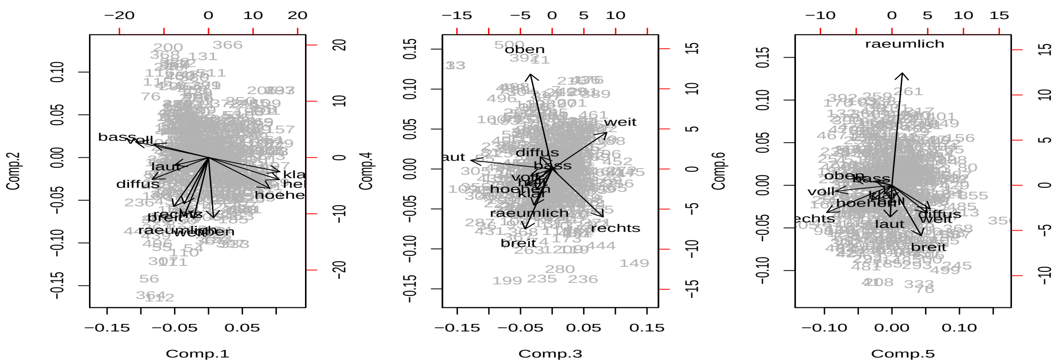


(b) Shape set

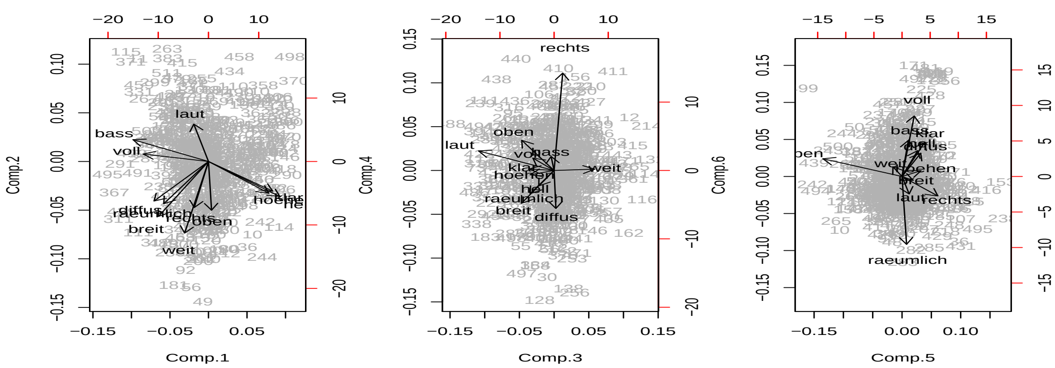


(c) Size set

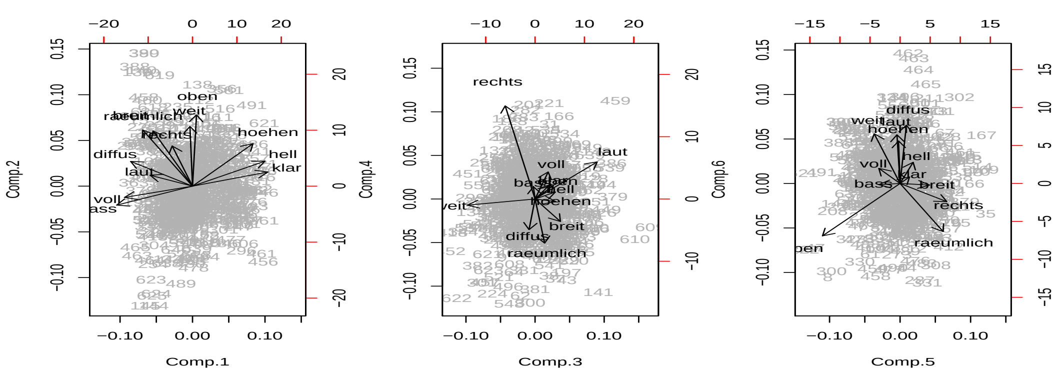
Figure 7.32: Scree plots for the principal component analyses.



(a) Refinement set



(b) Shape set



(c) Size set

Figure 7.33: Biplots for the principal component analysis, for each set.

Chapter 8

Conclusions

Wave Field Synthesis provides a solution to many limitations found in other spatialisation techniques where a convincing reproduction is only achieved in a 'sweet spot'. WFS excels at creating a clear localisation of virtual sound sources for a large listening area, by physically (re)creating the wave field. In order to do so, it does need a considerable investment in hardware: loudspeakers and computation power, and suffers from spatial aliasing. With the advent of manufacturers of loudspeaker panels, mostly containing 8 loudspeaker channels, and the increase in available computation power, even on the consumer market, the investment in a WFS system becomes affordable. While composers and sound artists alike are embracing the technology, considerable future developments would ensure that this becomes a viable sound presentation environment.

Since the first public WFS concert in 2003, there have been several works presented to audiences, and increasing numbers of composers are displaying an interest in working in this field. With more institutes investing in WFS technology, this in turn provides more opportunities for composers and sound artists to do so. Simultaneously, WFS research is also undergoing developments to enhance the possibilities of the technique beyond mere source movements. This includes methods for reproducing 3D sound objects that have been presented in this thesis.

As more composers become involved in using WFS this will increase demands on the possibilities that the software offers. These demands will include controlling parameters of the spatialisation, plus human computer interface models to do so, as well as demands for acoustical behaviour of the system. While WFS provides a means of recreating natural acoustic spaces, which has been the focus of many developments so far, composers interests lie beyond these representations. Their desires are also directed towards creating virtual and unnatural acoustic spaces. It could be surmised that the time has come to address not only the question of "How can we use Wave Field Synthesis?", but also "How can we abuse Wave Field Synthesis?".

While additional work can be directed towards improving the software as presented in chapter 3, future work should be focussed on establishing a compatible format between WFS systems as well as between different reproduction methods. Of special interest here is addressing the aspect of scaling compositions between large and small systems, which is both a technological and aesthetic issue.

This thesis presents a method for reproducing arbitrarily shaped sound sources, based on an additive point source method where point sources are positioned on the surface of the (virtual) object. This method includes calculation of diffraction of the sound, around the object itself, based on the diffraction model by Svensson et al. [SF99]. Implementation of this method has been discussed in detail, as well as the acoustical and perceptual effects achieved with it.

A listening experiment shows that the perceived effect influences both tone colouration (characterising the object), as well as spatial effects with several dimensions such as source width and localisation. Further listening experiments are required to evaluate the influence of variations in the spatial phase and frequency relationships between points on the object surface.

The diffraction model that was chosen does have a weakness in that it is a geometrical model, and for the wave model on which WFS is based, this poses a problem as some of the reproduction speakers may be in a shadow zone for certain points on the surface of the object in question; this means that at the reproduction array a second kind of diffraction occurs, as from the reproduction array sound from the speakers neighbouring the shadowed zone will enter this shadow zone. A solution could be a model which calculates the diffraction for a representative (receiver) area in the listening zone and does a backwards transformation to the loudspeaker array to calculate the reproduction signals.

Though the method has been presented for WFS reproduction, components of this method could be used for other purposes, such as binaural reproduction. For the binaural reproduction, the problem with the diffraction model will not be a problem, as the signals can be calculated directly for the location of the listeners' ears.

The method will be useful for many different applications, varying from artistic applications, to creating virtual environments, both in industrial applications (e.g. for auralisation of architecture before buildings are actually built) and training/education applications (e.g. military training, sound engineer training, visually impaired), as well as research applications.

In order to make full use of the method, it will have to be connected with physical modelling of the source object surface vibration.

In the development of the method presented the emphasis has been on attempting to create a method that produces an accurate reproduction, partially at the cost of computational complexity. It should be subject of future research in how far such accuracy is needed to achieve an effective, plausible reproduction, which is computationally simpler, and therefor capable of realtime application (without having to render in advance any possible source constellation that may occur).

Outlook

Wave field synthesis as audio reproduction method opens up a whole new range of possibilities, which were previously impossible. This can have a serious effect on how we think about what is a good audio reproduction: over 100 years of working with mono and stereophonic techniques have accustomed us to hearing originally acoustic music mainly through electro-acoustic reproduction methods, which involve recording the acoustical music, followed by various manipulations of the recording (including editing and changing the spatial relationships to sound "good" on stereophonic reproduction) by sound engineers, who follow their own aesthetics. Thus for many listeners the acoustic (real) event is no longer the reference to which they compare a recording, but rather they compare the acoustic event (in which musicians may make mistakes, or other nuances, which are usually lost in the editing process), when they visit a performance (in which they may not be seated in the ideal spot), to a polished recording. Future sound engineers, who produce recorded music for WFS, may choose different aesthetics: they may want to return to come closer to the acoustic event, or they may want to have more subtle means for polishing the recording, by having a larger control over all kinds of different parameters.

Another interesting point is that the clarity of localisation with WFS, sets higher demands on the accuracy of working with it: as sounds are more easily localised, and thus segregated, finer nuances in the sound will be heard. In other words, it is harder "to get away with" inaccuracies.

As spatial qualities will gain more importance in the creation process of content, this may eventually also train the listeners more in spatial hearing.

A merge of WFS technology with architecture will be necessary for the technology to become widespread, and come to the consumer's homes. This is not only interesting for reproducing entertainment content, but it also fits into concepts of sonic architecture, where a building is made suitable for living in, not only by the means of ergonomic structural design, choice of colours, lightning, etc., but also in an acoustical sense, which is then no longer limited to eliminating noise problems that are structurally transmitted, or improvement of the acoustics of a space, but also give options for enhancing the space with sound, which makes a person feel comfortable within the space.

WFS is an excellent technology for augmented reality, where virtual reality interacts with "real" reality. Whereas for visual virtual reality the problem is that the objects modelled have no substance, i.e. we cannot touch the virtual object, for acoustically modelled objects, we do not need a substance, as sounds do not normally have substance in nature; in other words, in order to interact with an auditory virtual reality the reproduced sound is enough: there is no need for further modelling through haptic devices. This makes it easier to merge an acoustic virtual reality with the real world and so create an augmented reality. WFS, especially when merged with architecture in such a way that its hardware is not recognisable, provides excellent methods to create such an augmented reality, as it can easily create a window into a virtual world, and even open this window, as virtual sounds can be positioned in front of the loudspeakers.

... with a focus on the reproduction of arbitrarily shaped sound sources

Acknowledgements

This Ph.D. research project was funded by a grant, provided by the Nachwuchsförderungsgesetz, *NaFöG*, in Berlin, Germany.

The listening tests were performed at the Fraunhofer Institute for Digital Media in Ilmenau, with the kind support of Judith Liebetrau and other employees of the institute, as well as students and scientists from the Technische Universität Ilmenau.

Furthermore, I would like to thank Torben Hohn, Simon Schampijer, Thilo Koch, Daniel Plewe, Eddie Mond, Wilm Thoben and Sebastian Roos for their input into the WFS system and technical support for the composers. I would also like to thank the composers and sound artists Kirsten Reese, Hans Tutschku, Christian Calon, Victor Lazzarini, André Bartetzki, Edwin van der Heide, Shintaro Imai and Wouter Snoei for their cooperation.

The Acoustics Toolbox for Matlab from the Sound Control Group of the University of Technology, Delft was used for the simulations in chapters 2 and 5.

This work would not have been possible without the following open source programs:

- Octave for analysis of the impulse responses in chapter 7, used in conjunction with PLplot and OctPlot[1] for data visualisation.
- SuperCollider for the listening tests, using SwingOSC[2] for the graphical interface.
- R[3] for statistical analysis of the listening test results,
- BruteFIR[4],
- Qt-libraries[5],
- KDevelop[6],
- LaTeX and Kile[7] for typesetting this thesis,
- plus the Debian, Agnula and 64studio GNU/Linux distributions[8].

[1] `http://www.octave.org`, `http://sourceforge.net/projects/plplot` and `http://octplot.sourceforge.net`

[2] `http://supercollider.sourceforge.net` and `http://www.sciss.de/swingOSC`

[3] `http://www.R-project.org`

[4] `http://www.ludd.luth.se/~torger/brutefir.html`

[5] `http://www.trolltech.com/products/qt/`

[6] `http://www.kdevelop.org`

[7] `http://kile.sourceforge.net`

[8] `http://www.debian.org`,`http://www.agnula.org` and `http://www.64studio.com`

Bibliography

[AB01] Juan Allende-Blin. *Der Lärm der Strasse - Italienischer Futurismus 1909-1918*, chapter Der Italienische Futurismus in der Musik. Norbert Nobis, Sprengel Museum Hannover, 2001.

[Adr91] J.M. Adrien. *The Missing Link: Modal Synthesis, in Representations of Musical Signals*. MIT Press, 1991. pages: 269-297.

[AdV04] Bernhard Albrecht and Diemer de Vries. Multichannel Musikaufnahmen mit kreisförmigem Mikrophonarray. In *23. Tonmeistertagung, Leipzig*, November 2004.

[Agr76] B.N. Agrawal. Modal synthesis technique for dynamic analysis of structures. *Journal of the Acoustical Society of America*, 59(6):1329–1338, June 1976.

[And85] Y. Ando. *Concert Hall Acoustics*. Springer Series in Electrophysics 17, Berlin, 1985.

[B9̈3] Konrad Böhmer. *Das böse Ohr*, chapter Raum-Former (1961). Dumont Dok., 1993.

[Baa03] M.A.J. Baalman. Application of wave field synthesis in the composition of electronic music. In *International Computer Music Conference 2003, Singapore, 29 September - 4 October 2003*, 2003.

[Baa04] M.A.J. Baalman. Application of wave field synthesis in electronic music and sound installations. In *2nd International Linux Audio Conference, 29 april - 2 Mai 2004, ZKM, Karlsruhe*, 2004.

[Baa05] M.A.J. Baalman. Updates of the WONDER software interface for using wave field synthesis. In *3rd International Linux Audio Conference, April 21-24, 2005, ZKM, Karlsruhe*, 2005.

[BAM07] T. Behrens, W. Ahnert, and C. Moldrzyk. Raumakustische Konzeption von Wiedergaberäumen für Wellenfeldsynthese am Beispiel eines Hörsaals der TU Berlin. In *DAGA 2007, Stuttgart, Germany*, 2007.

[Bar02] Natasha Barrett. Spatio-musical composition strategies. *Organised Sound, Cambridge University Press*, 7(3):313–323, 2002.

[Bar07] André Bartetzki. A software based mixing desk for acousmatic sound diffusion. Presentation on the Linux Audio Conference 2007, Berlin, Germany, 2007.

[BB06] Michael Bender and Manfred Brill. *ComputerGrafik*. Number ISBN 3-446-46434-1. Hanser, 2. auflage edition, 2006.

[BCJ01] N. Bleistein, J.K. Cohen, and J.W. Stockwell Jr. *Mathematics of multidimensional seismic imaging, migration, and inversion*. Springer-Verlag, New York, 2001.

[Ber88] A.J. Berkhout. A holographic approach to acoustic control. *Journal of the Audio Engineering Society*, 36(12):977–995, December 1988.

[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. In *Linux Audio Conference 2007, TU Berlin, 22-25 March 2007*, 2007.

[Bil06] S. Bilbao. Fast modal synthesis by digital waveguide extraction. *Signal Processing Letters, IEEE*, 13(1):1–4, January 2006.

[Bis00] I. Bisnovatyi. Flexible software framework for modal synthesis. In *Conference on Digital Audio Effects (DAFX-00), Verona, Italy*, December, 7-9 2000.

[BK03] I. Bork and M. Kern. Simulation der schallabstrahlung eines flügels. In *DAGA '03*, 2003.

[Bla97] J. Blauert. *Spatial Hearing*. MIT Press, revised edition edition, 1997.

[Ble84] N. Bleistein. *Mathematical Methods for Wave Phenomena*. Academic Press, New York, 1984.

[BM04] Mark J. Beeson and Damian T. Murphy. Roomweaver: a digital waveguide mesh based room acoustics research tool. In *Proceedings of the yth International Conference on Digital Audio Effects (DAFx '04), Naples, Italy*, October 2004.

[Boe92] Jan Boerman. Muziek en ruimte. Program booklet of the Rumori series 1991/92, Frascati, Amsterdam, 1992.

[Bor26] E.G. Boring. Auditory theory with special reference to intensity, volume and localisation. *Journal of the Acoustical Society of America*, 37(2):157–188, 1926.

[Bor05] Jürgen Bortz. *Statistik für Human- und Sozialwissenschaftler.* Number ISBN 3-540-21271-X. Springer Medizin Verlag Heidelberg, 6. auflage edition, 2005.

[BP04] M.A.J. Baalman and D. Plewe. WONDER - a software interface for the application of wave field synthesis in electronic music and interactive sound installations. In *International Computer Music Conference 2004, Miami, 1-6 November 2004*, 2004.

[BR99] Jan Berg and Francis Rumsey. Spatial attribute identification and scaling by repertory grid technique and other methods. In *AES 16th International Conference*, 1999.

[BT57] M.A. Biot and I. Tolstoy. Formulation of a wave propagation in infinite media by normal coordinates with an application to diffraction. *Journal of the Acoustical Society of America*, 29:381–391, 1957.

[Cag67] John Cage. *A Year from Monday*, chapter Diary: Audience 1966, pages 50–51. Middetown, Connecticut: Wesleyan University Press, 1967.

[Can05] Helmut Cantzler. Mesh viewer. `http://mview.sourceforge.net/`, 2001-2005.

[CCW03] T. Caulkins, E. Corteel, and O. Warusfel. Wave field synthesis interaction with the listening environment, improvements in the reproduction of virtual sources located inside the listening room. In *Proceedings of the 6th International Conference on Digital Audio Effects (DAFx '03), London, UK*, September 2003.

[Cho71] J. Chowning. The simulation of moving sound sources. *Journal of the Audio Engineering Society*, 19(1):2–6, 1971. Reprinted in *Computer Music Journal*, vol. 1, no. 3, pp. 48-52.

[CHP02] E. Corteel, U. Horbach, and R. Pellegrini. Multichannel inverse filtering of multiexciter distributed mode loudspeakers for wave field synthesis. In *112th convention of the Audio Engineering Society, Munich,*, 2002.

[Clo01] Christian Clozier. The gmebaphone concept and the cybernèphone instrument. *Computer Music Journal*, 25(4):81–90, Winter 2001.

[Cor06] E. Corteel. Equalization in extended equalization in an extended area using multichannel inversion and wave field synthesis. *Journal of the audio Engineering Society*, 54(12), December 2006.

[Cor07] E. Corteel. Synthesis of directional sources using wave field synthesis, possibilities, and limitations. *EURASIP Journal on Advances in Signal Processing*, 2007.

[CS05] Paul T. Calamia and U. Peter Svensson. Edge subdivision for fast diffraction calculations. In *2005 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*. New Paltz, NY, October 16-19 2005.

[CS07] Paul T. Calamia and U. Peter Svensson. Fast time-domain edge-diffraction calculations for interactive acoustic simulations. *EURASIP Journal on Advances in Signal Processing*, 2007:Article ID 63560, 10 pages, 2007. doi:10.1155/2007/63560.

[CSF05] Paul T. Calamia, U. Peter Svensson, and Thomas A. Funkhouser. Integration of edge-diffraction calculations and geometrical-acoustics modeling. In *Forum Acusticum 2005 Budapest*, 2005.

[Dea] Paul Davis and et al. Jack audio connection kit. `http://jackaudio.org/`.

[dlMH99] Helga de la Motte-Haber, editor. *Klangkunst - Tönende Objekte und klingende Räume*, volume 12 of *Handbuch der Musik im 20. Jahrhundert*. Laaber-Verlag, Laaber, Germany, 1999.

[DNM03] Jérôme Daniel, Rozenn Nicol, and Sébastien Moreau. Further investigations of high order ambisonics and wavefield synthesis for holophonic sound imaging. In *Proceedings of the 114th Convention of the Audio Engineering Society, Amsterdam, the Netherlands*, March 22-25 2003.

[dVHG07] Diemer de Vries, Lars Hörchens, and Peter Grond. Extraction of 3D information from Circular Array Measurements for Auralization with Wave Field Synthesis. *EURASIP Journal on Advances in Signal Processing*, 2007.

[dVLM06] Diemer de Vries, Jan Langhammer, and Frank Melchior. A new approach for direct interaction with graphical representations of room impulse responses for the use in wave field synthesis reproduction. In *120th AES Convention, Paris, Preprint 6657*, May 2006.

[dVvDSvdH07] Diemer de Vries, Jasper van Dorp Schuitman, and At van den Heuvel. A new digital module for variable acoustics and wave field synthesis: Design and applications. In *AES 122nd Convention, Vienna, Austria*, 2007.

[Eat02] John W. Eaton. *GNU Octave Manual*. Network Theory Limited, 2002.

[EIC95] G. Eckel, F. Iovino, and R. Caussé. Sound synthesis by physical modeling with modalys. *International Symposium of Music Acoustics, Le Normant, Dourdan*, 1995.

[FAWK06] A. Freed, R. Avizienis, D. Wessel, and P. Kassakian. A compact 120 independent element spherical loudspeaker array with programmable radiation patterns. In *AES 120th Convention, Paris, France*, 2006.

[FBA03] Hania Farag, Jens Blauert, and Onsy Abdel Alim. Psychoacoustic investigations on sound-source occlusion. In *AES 114th Convention, Amsterdam, The Netherlands*, 2003.

[Fen06] John D. Fenton. Numerical methods. `http://www.ifh.uni-karlsruhe.de/people/fenton/LectureNotes/Numerical-Methods.pdf`, Institut für Hydromechanik, Universität Karlsruhe, Germany, November 2006.

[FJ05] Matteo Frigo and Steven G. Johnson. The design and implementation of FFTW3. *Proceedings of the IEEE*, 93(2):216–231, 2005. special issue on "Program Generation, Optimization, and Platform Adaptation".

[FR98] Neville H. Fletcher and Thomas D. Rossing. *The Physics of Musical Instruments.* Springer Science + Business Media, Inc., 2nd edition, 1998.

[Fur] Richard Furse. Linux Audio Developer's Simple Plugin API (LADSPA). `http://www.ladspa.org`.

[GGF96] Frank Gertich, Julia Gerlach, and Golo Föllmer. *Musik..., verwandelt. Das Elektronische Studio der TU Berlin 1953-1995.* Wolke Verlag, 1996.

[GMMW07] A. Goertz, M. Makarski, C. Moldrzyk, and S. Weinzierl. Entwicklung eines achtkanaligen Lautsprechermoduls für die Wellenfeldsynthese. In *DAGA 2007, Stuttgart, Germany*, 2007.

[Gre00] M.L. Green. *GEOGRAPHIC INFORMATION SYSTEM BASED MODELING OF SEMI-VOLATILE ORGANIC COMPOUNDS TEMPORAL AND SPATIAL VARIABILITY.* PhD thesis, University of New York at Buffalo, 2000.

[gts05] Gnu triangulated surface library. `http://gts.sourceforge.net/`, 2000-2005.

[HdV01] Edo Hulsebos and Diemer de Vries. Spatial decomposition and data reduction of sound fields measured using microphone array technology. In *17th ICA, Rome, 2001*, 2001.

[Hel03] Hiske Helleman. Sensitivity of the human auditory system to spatial variations in single early reflections. Master's thesis, Delft University of Technology, The Netherlands, 2003.

[Hoe07] Michael Hoeldke. Der Kanon als räumliche Abbildung bei Nancarrow. In *next generation: "Musik im Raum", Symposium.* Zentrum für Kunst und Medien (ZKM), Karlsruhe, Germany, June 2007.

[HSdVB03] Edo Hulsebos, Thomas Schuurmans, Diemer de Vries, and Rinus Boone. Circular microphone array for discrete multichannel audio recording. In *114th AES Convention, Amsterdam, Preprint 5716*, March 2003.

[Huy90] Christiaan Huygens. *Traite de la lumiere; ou sont expliquees les causes de ce qui luy arrive dans la reflexion et dans la refraction et particulierement dans l'etrange refraction du cristal d'Islande; avec un discours de la cause de la pesanteur.* P. van der Aa, Leiden, The Netherlands, 1690.

[HvdDF03] Reynald Hoskinson, Kees van den Doel, and Sidney Fels. Realtime adaptive control of modal synthesis. In *2003 International Conference on New Interfaces for Musical Expression, McGill University, Montreal, Canada 22-24 May 2003, Proceedings*, pages 99–103, May 2003.

[IRC03] IRCAM. *Modalys Reference Manual - Finite Element Objects*, September 2003.

[IRC05] IRCAM. Modalys. `http://www.ircam.fr/`, 1991-2005.

[Itu07] Manuel Rocha Iturbide. The sound installation. In *International Computer Music Conference 2007, Copenhagen, Danmark, August 27-31, 2007*, volume 2, pages 17–20, 2007.

[JAMdV05] R. Jacques, B. Albrecht, F. Melchior, and D. de Vries. An approach for multichannel recording and reproduction of sound source directivity. In *Proceedings of the 119th Convention of the Audio Engineering Society (AES '05), New York, NY, USA*, October 2005.

[Jan97] G. Jansen. Focused wavefields and moving virtual sources by wavefield synthesis. Master's thesis, University of Technology Delft, The Netherlands, August 1997.

[Joh87] M. Johnson. *The Body in the Mind: The Bodily Basis of Meaning, Imagination and Reason.* University of Chicago, 1987.

[Kel62] J.B. Keller. Geometrical theory of diffraction. *Journal of the Optical Society of America*, 52(2):116–130, 1962.

[Ken95] Gary S. Kendall. The decorrelation of audio signals and its impact on spatial imagery. *Computer Music Journal*, 19(4):71–87, Winter 1995.

[Ken07] Gary S. Kendall. The artistic play of spatial organization: spatial attributes, scene analysis and auditory spatial schemata. In *International Computer Music Conference 2007, Copenhagen, Danmark, August 27-31, 2007*, volume 1, pages 63–68, 2007.

[Ker06] Stefan Kersten. A fast convolution engine for the virtual electronic poem project. Master's thesis, Technische Universität Berlin, Berlin, Germany, 2006.

[KMP04] M. Karjalainen and T. Mäki-Patola. Physics-based modeling of musical instruments for interactive virtual reality. 2004.

[Kop05] Stephan Kopp. Programmierung eines Plugins zur Dekodierung quadrofon matrizierter Audiosignale. Master's thesis, Technische Universität Berlin, Berlin, Germany, April 2005.

[KP74] R.G. Kouyoumjian and P.H. Pathak. A uniform geometrical theory of diffraction for an edge in a perfectly conducting surface. *Proceedings of the IEEE*, 62(11):1448–1461, 1974.

[Kuh06] Clemens Kuhn. Article about the sonic emotion's installation in Centre Pompidou, Paris for the Dada exhibition, in "Technik on Stage" product guide for events. `http://www.sonicemotion.com/ch/images/PDF/sonic_emotion_bericht_tos_2006_clean.pdf`, 2006.

[Mat] The Mathworks. Matlab. `http://www.mathworks.com`.

[Max] Max/MSP programming environment. `http://www.cycling74.com/products/maxmsp.html`.

[MB88] Mariette Morawski-Büngeler. *Schwingende Elektronen. Eine Dokumentation über das Studio für Elektronische Musik des Westdeutschen Rundfunks in Köln 1951-1986.* Köln-Rodenkirchen, 1988.

[McC] J. McCartney. Supercollider. `http://www.audiosynth.com`.

[MCJ82] H. Medwin, E. Childs, and G.M. Jebsen. Impulse studies of double diffraction: A discrete huygens interpretation. *Journal of the Acoustical Society of America*, 72:1005–1013, 1982.

[McM06] Alex McMahon. Supermono: Development of a spectral composition and a real-time wavefield synthesis system. Master's thesis, Music & Media Technologies, Department of Electronic & Electrical Engineering & School of Music, Trinity College of Dublin, Ireland, 2006.

[MdV06] Frank Melchior and Diemer de Vries. Detection and visualization of early reflections for wave field synthesis sound design applications. In *Tonmeistertagung 2006, Leipzig, Germany*, November 16-19 2006.

[Mey95] J. Meyer. *Akustik und musikalische Aufführungspraxis.* Das Musikinstrument, Frankfurt/M, 1995.

[MGM+07] C. Moldrzyk, A. Goertz, M. Makarski, W. Ahnert, S. Feistel, and S. Weinzierl. Wellenfeldsynthese für einen großen Hörsaal. In *DAGA 2007, Stuttgart, Germany*, 2007.

[MM95] D.G. Malham and A. Myatt. 3-d sound spatialization using ambisonic techniques. *Computer Music Journal*, 19(4):58–70, 1995.

[Mon07] Hans-Joachim Mond. Implementierung einer graphischen Benutzeroberfläche zur Steuerung eines Wellenfeldsynthesesystems. Master's thesis, Technische Universität Berlin, Berlin, Germany, July 2007.

[Moo04] Brian C.J. Moore. *An introduction to the psychology of hearing.* Elsevier Academic Press, 2004.

[Mor02] M. Morimoto. The relationship bewteen spatial impression and the precedence effect. In *Proc. of the ICAD 2002 Conference, Kyoto, Japan*, July 2002.

[MW03] Ali Momeni and David Wessel. Characterizing and controlling musical material intuitively with geometric models. In *2003 International Conference on New Interfaces for Musical Expression, McGill University, Montreal, Canada 22-24 May 2003, Proceedings*, pages 54–61, 2003.

[MWC01] N. Misdariis, O. Warusfel, and R. Caussé. Radiation control on a multi-loudspeaker device. In *ISMA 2001*, 2001.

[PB82] D.R. Perrott and T.N. Buell. Judgements of sound volume: effects of signal duration, level, and interaural characteristics on the perceived extensity of broadband noise. *Journal of the Acoustical Society of America*, 72(5):1413–1417, November 1982.

[PB03] Guillaume Potard and Ian Burnett. A study on sound source apparent shape and wideness. In *ICAD 2003*, 2003.

[PBG03] J. Plogsties, O. Baum, and B. Grill. Conveying spatial sound using MPEG-4. In *AES 24th Conference, Banff, Canada*, 2003.

[Pot06] Guillaume Potard. *3D-Audio Object Oriented Coding*. PhD thesis, University of Wollongong, Australia, September 2006.

[R D07] R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2007. ISBN 3-900051-07-0.

[Raa93] Dick Raaijmakers. Gecomponeerde ruimte: van interactie in de hedendaagse muziek tot holofonische ruimtelijkheid. 1993.

[RBPL05] Emeline Reboul, Alain Le Bot, and Joël Perret-Liaudet. Radiative transfer equation for multiple diffraction. *Journal of the Acoustical Society of America*, 118(3):1326–1334, September 2005.

[RGB06] Chandrasekhar Ramakrishnan, Joachim Goßmann, and Ludger Brümmer. The ZKM klangdom. In Michael Lyons Norbert Schnell, Frédéric Bevilacqua and Atau Tanaka, editors, *New Interfaces for Musical Expression - NIME 06*, pages 140–143. IRCAM - Centre Pompidou in collaboration with MINT/OMF, Sorbonne University, 2006.

[RME] RME - intelligent audio solutions. `http://www.rme-audio.com/`.

[Rod05] René Rodigast. *Tagungsband Festsymposium 20 Jahre Fraunhofer IIS - Anwendungen der Mikroelektronik in der Informations- und Kommunikationstechnik, 1.7.2005*, chapter Neuer Sound für die Bregenzer Festspiele - Innovative Sound for the Bregenz Festival, pages 54 – 55. Heinz Gerhäuser, 2005.

[Rum02] F. Rumsey. Spatial quality evaluation for reproduced sound: Terminology, meaning and a scene-based paradigm. *Journal of the Audio Engineering Society (JAES)*, 50(9), 2002.

[Rut] Hanns Holger Rutz. SwingOSC. `http://www.sciss.de/swingOSC`.

[Rut04] Hanns Holger Rutz. Meloncillo - eine graphische benutzeroberfläche zur musikalischen raumklangsteuerung mit imlementierung einer OSC-schnittstelle zur klangsynthese. Master's thesis, Technische Universität Berlin, Berlin, Germany, 2004.

[Sav00] Lauri Savioja. *Modeling Techniques for Virtual Acoustics*. PhD thesis, Helsinki University of Technology, August 2000.

[SC06] U. Peter Svensson and Paul T. Calamia. Edge-diffraction impulse responses near specular-zone and shadow-zone boundaries. *Acta Acustica united with Acustica*, 92(4):501–512(12), July/August 2006.

[Sch90] Pierre Schaeffer. Journal d'orphée. In F. Bayle, editor, *Pierre Schaeffer l'œuvre musicale*. Paris: INA/GRM and Librairie SEGUIER, 1990.

[SdV97] J.J. Sonke and D. de Vries. Generation of diffuse reverberation by plane wave synthesis. In *102nd AES Convention, March 1997, Preprint 4455*, 1997.

[SF99] U.P. Svensson and R.I. Fred. An analytic secondary source model of edge diffraction impulse responses. *Journal of the Acoustical Society of America*, 106(5):2331–2344, November 1999.

[Smi92] J.O. Smith. Physical modelling using digital waveguides. *Computer Music Journal*, 16(4):74–87, Winter 1992.

[Smi97] Steven W. Smith. *The Scientist and Engineer's Guide to Digital Signal Processing*. California Technical Publishing, 1997.

[Sno06] Wouter Snoei. Wave Field Synthesis with the Game of Life. `http://www.woutersnoei.nl/?c=welcome&s=archief&l=en`, `http://www.woutersnoei.nl/hku/`, 2006.

[Som89] P.C.W. Sommen. Partitioned frequency domain adaptive filters. In *Proceedings Asilomar Conference on Signals, Systems and Computers*, pages 676–681, 1989.

[Son00] J.J. Sonke. *Variable Acoustics by Wave Field Synthesis*. PhD thesis, University of Technology, Delft, 2000.

[Spo05] Sascha Spors. *Active Listening Room Compensation for Spatial Sound Reproduction Systems*. PhD thesis, Friedrich-Alexander-University, Erlangen-Nürnberg, 2005.

[Sta97] Evert W. Start. *Direct sound enhancement by wave field synthesis*. PhD thesis, University of Technology, The Netherlands, 1997.

[Ste04] Uwe Martin Stephenson. *Beugungssimulation ohne Rechenzeitexplosion: Die Methode der quantisierten Pyramidenstrahlen - ein neues Berechnungsverfahren für Raumakustik und Lärmimmisionsprognose. Vergleiche, Ansätze, Lösungen*. PhD thesis, Rheinisch-Westfalische Technische Hochschule Aachen, July 2004.

[Ste06] Sternenrest (Sterren maken geen muziek). `http://www.fenomenologie.nl/Sternenrest/Sternenrest-Concept-Website.pdf`, 2006.

[Tec05] Geometry Technologies. Geomview. `http://www.geomview.org/`, 1992-2005.

[Tor05] A. Torger. Brutefir. `http://www.ludd.luth.se/~torger/brutefir.html`, 2001-2005.

[Tro01] Maja Trochimczyk. From circles to nets: on the signification of spatial sound imagery in new music. *Computer Music Journal*, 25(4):39–56, Winter 2001.

[Tro05] Trolltech. Qt library. `http://www.trolltech.com/products/qt/`, 1996-2005.

[TWR02] G. Theile, H. Wittek, and M. Reisinger. Wellenfeld-Synthese Verfahren: Ein Weg für neue Möglichkeiten der räumlichen Tongestaltung. In *22nd Tonmeistertagung, Hannover, Germany, 2002 November*, 2002.

[Vag01] Horacio Vaggione. Composing musical spaces by means of decorrelation of audio signals. In *Addendum of the COST G-6 Conference on Digital Audio Effects (DAFX-01), Limerick, Ireland*, pages Addendum–1–8, December 6-8 2001.

[Van91] John Vanderkooy. A simple theory of cabinet edge diffraction. *Journal of the Audio Engineering Society (JAES)*, 31(12):923–933, December 1991.

[Ver98] E.N.G. Verheijen. *Sound Reproduction by Wave Field Synthesis*. PhD thesis, University of Technology, The Netherlands, 1998.

[VST] Virtual Studio Technology (VST). `http://en.wikipedia.org/wiki/Virtual_Studio_Technology`, `http://ygrabit.steinberg.de`.

[WCMC04] O. Warusfel, E. Corteel, N. Misdariis, and T. Caulkins. Reproduction of sound source directivity for future audio applications. In *ICA 2004*, 2004.

[Weg05] Dominik Wegmann. Zu Unterschieden in der Hörereigniswahrnehmung bei Wellenfeldsynthese und Stereofonie im Vergleich zum natürlichen Hören. Master's thesis, Fachhochschule Oldenburg / Ostfriesland / Wilhelmshafen, 2005. `http://www.hauptmikrofon.de`.

[Wei03] E.W. Weisstein. *CRC Concise Encyclopedia of Mathematics*. Chapman & Hall/CRC, `http://mathworld.wolfram.com`, 2003.

[Wei05a] Eric W. Weisstein. Simpson's rule. From MathWorld–A Wolfram Web Resource. `http://mathworld.wolfram.com/SimpsonsRule.html`, April 2005.

[Wei05b] Eric W. Weisstein. Trilinear coordinates. From MathWorld–A Wolfram Web Resource. `http://mathworld.wolfram.com/TrilinearCoordinates.html`, December 2005.

[Wes01] Jörg Weske. Aufbau eines 24-kanal basissystems zur wellenfeldsynthese mit der zielsetzung der positionierung virtueller schallquellen im abhörraum. Master's thesis, Technische Universität Chemnitz, 2001.

[WFM03] M. Wright, A. Freed, and A. Momeni. Opensoundcontrol: State of the art 2003. In *2003 International Conference on New Interfaces for Musical Expression, McGill University, Montreal, Canada 22-24 May 2003, Proceedings*, pages 153–160, 2003.

[Wik06] Wikipedia. `http://en.wikipedia.org/wiki/Granular_synthesis`, 28 September 2006.

[Wik07] Wikipedia. Bresenhams line algorithm. `http://en.wikipedia.org/wiki/Bresenham's_line_algorithm`, 2007.

[Wit04] Helmut Wittek. Bericht zum Forum WFS der Oper Stuttgart. *VDT - Magazin*, Juni 2004.

[WM01] O. Warusfel and N. Misdariis. Directivity synthesis with a 3d array of loudspeakers application for stage performance. In *Proceedings of the COST G-6 Conference on Digital Audio Effects (DAFX-01), Limerick, Ireland, December 6-8*, 2001.

[WM04] O. Warusfel and N. Misdariis. Sound source radiation synthesis: from stage performance to domestic rendering. In *AES 116th Convention, Berlin, Germany, 2004, May, Preprint 6018*, 2004.

[WT06] Stefan Weinzierl and Kees Tazelaar. Raumsimulation und Klangkunst - vom Künstlichen Nachhall zur virtuellen Akustik. In Helga de la Motte-Haber, editor, *Sonambiente Berlin 2006 - Klangkunst - Sound Art*, pages 351–365. Helga de la Motte-Haber, Matthias Osterwold and Georg Weckwerth im Auftrag der Akademie der Künste, Kehrer Verlag Heidelberg, `www.sonambiente.net`, 2006.

[Xen94] I. Xenakis. Musique, espace, et spatialisation: Entretien de iannis xenakis avec maria harley. *Circuit, Revue Nord-Americaine de Musique du XXe Siècle*, 5(2):9–20, 1994.

Appendix A

XML file formats

```
 <?xml version="1.0"?>
<!DOCTYPE globalarray PUBLIC "" "globalarray.dtd">

<globalarray>
    <speakers distance="0.1" reference="15.0" />
    <focus limit="12.0" margin="3.0" />
    <elevation y1="8.8" y2="25.7" z1="2.1" z2="6.5" />
    <polygon>
        <point x="-7.7" y="0" z="2.1" />
        <point x="-10.0" y="8.8" z="2.1" />
        <point x="-10.0" y="25.7" z="6.5" />
        <point x="10.0" y="25.7" z="6.5" />
        <point x="10.0" y="8.8" z="2.1" />
        <point x="7.7" y="0" z="2.1" />
    </polygon>
</globalarray>
```

Figure A.1: Example of a global array configuration.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE array PUBLIC "" "array.dtd">

<array>
 &lt;segment numspeak="8" winwidth="0"
    startx="-0.402500" starty="0.020000" startz="2.115423"
    endx="0.302500" endy="0.020000" endz="2.115377"
    normx="-0.000000" normy="1.000000" normz="0.000000" />
 &lt;segment numspeak="8" winwidth="0"
    startx="0.406188" starty="0.020000" startz="2.115370"
    endx="1.111188" endy="0.020000" endz="2.115324"
    normx="-0.000000" normy="1.000000" normz="0.000000" />
 &lt;segment numspeak="8" winwidth="0"
    startx="1.214875" starty="0.020000" startz="2.115318"
    endx="1.919875" endy="0.020000" endz="2.115272"
    normx="-0.000000" normy="1.000000" normz="0.000000" />
 &lt;segment numspeak="8" winwidth="0"
    startx="2.023562" starty="0.020000" startz="2.115265"
    endx="2.728562" endy="0.020000" endz="2.115219"
    normx="-0.000000" normy="1.000000" normz="0.000000" />
 &lt;segment numspeak="8" winwidth="0"
    startx="2.832250" starty="0.020000" startz="2.115212"
    endx="3.537250" endy="0.020000" endz="2.115167"
    normx="-0.000000" normy="1.000000" normz="0.000000" />
 &lt;segment numspeak="8" winwidth="0"
    startx="3.640937" starty="0.020000" startz="2.115160"
    endx="4.345937" endy="0.020000" endz="2.115114"
    normx="-0.000000" normy="1.000000" normz="0.000000" />
 &lt;segment numspeak="8" winwidth="0"
    startx="4.449624" starty="0.020000" startz="2.115107"
    endx="5.154624" endy="0.020000" endz="2.115061"
    normx="-0.000000" normy="1.000000" normz="0.000000" />
</array>
```


Figure A.2: Example of an array configuration file, containing several linear segments.

```
<?xml version="1.0"?>
<!DOCTYPE swonderconfig PUBLIC "" "swonderconfig.dtd">

<swonderconfig>
  <cwaddr host="127.0.0.1" port="58100"/>
  <defaults projectpath="/home/wfsuser/swonder/projects" maxsources="64" />
</swonderconfig>
```

Figure A.3: Example of a *sWONDER* default configuration file, containing the IP address and port of *cwonder*, a default project path and maximum number of sources.

```
<?xml version="1.0"?>
<!DOCTYPE swonder SYSTEM "swonder.dtd">

<swonder name="example_project" folder="" samplerate="44100"
      arrayfile="" gridfile="">
  <scene id="0" name="demo">
    <source id="0" type="1" name="Source #1" mute="0"
      posx="-1.66187" posy="-4.41185" posz="1" angle="-0"/>
    <source id="1" type="0" name="Source #2" mute="0"
      posx="-11.1" posy="-2.65" posz="1" angle="37"/>
    <source id="2" type="1" name="Source #3" mute="0"
      posx="10.95" posy="4.75" posz="1" angle="-0"/>
  </scene>
  <scene id="1" name="demo2">
    <source id="0" type="1" name="Source #1" mute="0"
      posx="-14.8619" posy="7.93815" posz="1" angle="-0"/>
    <source id="1" type="0" name="Source #2" mute="0"
      posx="-14.05" posy="12" posz="1" angle="37"/>
    <source id="2" type="1" name="Source #3" mute="0"
      posx="15.1" posy="16.95" posz="1" angle="-0"/>
  </scene>
  <scene id="2" name="demo3">
    <source id="0" type="1" name="Source #1" mute="0"
      posx="-1.66187" posy="-4.41185" posz="1" angle="-0"/>
    <source id="1" type="0" name="Source #2" mute="0"
      posx="-11.1" posy="-2.65" posz="1" angle="37"/>
    <source id="2" type="1" name="Source #3" mute="0"
      posx="10.95" posy="4.75" posz="1" angle="-0"/>
  </scene>
</swonder>
```

Figure A.4: Example of a *sWONDER* project file, containing three scenes.

```
<?xml version="1.0"?>
<!DOCTYPE scorefile SYSTEM "score.dtd">
<scorefile date="2007-09-08 15:49:19">
  <orchestra>
    <src id="0" type="1" posx="0.153824" posy="0.0034" posz="0.0045"
        rotx="0" roty="0" rotz="0" angle="0"/>
    <src id="1" type="1" posx="0.153847" posy="0.0034" posz="0.0045"
        rotx="0" roty="0" rotz="0" angle="0"/>
    <src id="2" type="1" posx="0.15387" posy="0.0034" posz="0.0045"
        rotx="0" roty="0" rotz="0" angle="0"/>
    <src id="3" type="1" posx="0.153893" posy="0.0034" posz="0.0045"
        rotx="0" roty="0" rotz="0" angle="0"/>
    <src id="4" type="1" posx="0.153916" posy="0.0034" posz="0.0045"
        rotx="0" roty="0" rotz="0" angle="0"/>
    <src id="5" type="1" posx="0.153939" posy="0.0034" posz="0.0045"
        rotx="0" roty="0" rotz="0" angle="0"/>
    <src id="6" type="1" posx="0.153962" posy="0.0034" posz="0.0045"
        rotx="0" roty="0" rotz="0" angle="0"/>
    <src id="7" type="1" posx="0.153985" posy="0.0034" posz="0.0045"
        rotx="0" roty="0" rotz="0" angle="0"/>
    <src id="8" type="1" posx="0.154008" posy="0.0034" posz="0.0045"
        rotx="0" roty="0" rotz="0" angle="0"/>
    <src id="9" type="1" posx="0.154031" posy="0.0034" posz="0.0045"
        rotx="0" roty="0" rotz="0" angle="0"/>
    <src id="10" type="1" posx="0.154054" posy="0.0034" posz="0.0045"
        rotx="0" roty="0" rotz="0" angle="0"/>
  </orchestra>
  <score>
    <pos t="9.35" dur="0" id="0" x="0" y="0.0034" z="0.0045"/>
    <pos t="9.36" dur="0" id="1" x="2.3e-05" y="0.0034" z="0.0045"/>
    <pos t="9.38" dur="0" id="2" x="4.6e-05" y="0.0034" z="0.0045"/>
    <pos t="9.39" dur="0" id="3" x="6.9e-05" y="0.0034" z="0.0045"/>
    <pos t="9.4" dur="0" id="4" x="9.2e-05" y="0.0034" z="0.0045"/>
    <pos t="9.41" dur="0" id="5" x="0.000115" y="0.0034" z="0.0045"/>
    <pos t="9.42" dur="0" id="6" x="0.000138" y="0.0034" z="0.0045"/>
    <pos t="9.43" dur="0" id="7" x="0.000161" y="0.0034" z="0.0045"/>
    <pos t="9.45" dur="0" id="8" x="0.000184" y="0.0034" z="0.0045"/>
    <pos t="9.46" dur="0" id="9" x="0.000207" y="0.0034" z="0.0045"/>
    <pos t="9.47" dur="0" id="10" x="0.00023" y="0.0034" z="0.0045"/>
  </score>
</scorefile>
```


Figure A.5: Example of a score file.

Appendix B

Program notes

B.1 Concert Program Club Transmediale 2003

Tuesday, February, 4th, 2003

This concert is supported by the Electro-Acoustic Studio of the TU Berlin. Furthermore, this concert would not have been possible without the program BruteFIR of Anders Torger.

Presentation about Wave Field Synthesis

Marije Baalman
Wave Field Synthesis gives the possibility to construct a wavefield as desired, which means that one can arbitrarily place a virtual sound source in space and with Wave Field Synthesis, one can calculate appropriate loudspeaker signals to create a wave field as it would be when a real sound source would have been at the specified location. At the moment the necessary calculation capacity make it possible to let such a system run in realtime and enable the virtual sound sources to move through space. The system has as an advantage above other systems that the effect is not based on psycho-acoustic theories and has a large listening area; this makes the technique ideal for concert environments.
The principles of the system will be presented, as well as the software written to enable composers to make use of the system.
More information can be found at: `http://www.kgw.tu-berlin.de/~baalman/`

Bunte Flügel

Frieder Butzmann - (1974/2003) - 6:19
1974. Rejected by the girl friend. What else could I do than making more electronic music?! I constructed and soldered, together with a friend, the machine "Duo Multivibratorsimiltan Hick Hack". With this, I recorded many long tones, flying high and falling even deeper, and distributed them over four channels. The tones had to fly! (Ja, die Liebe hat bunte Flügel. Solch einen Vogel fängt man schwer" / Carmen, Bizet). But they wouldn't fly. Maybe they fly

now with WFS.

hermetische_garage

Boris Hegenbart - (2003) - 5:57
is a track from the new [#/TAU]-release
`[somethingmovinginsideplasticbox]`
`[smip]` will be released in spring 2003 on Quecksilber/Staubgold

Restored to Life

Ilka Theurich - (2003) - 10:20
'Since 1999, I am experimenting with sound and more and more SOUNDart-PERFORMANCES come to live. At the moment the central focus of my artistic work is at a performative soundgeneration and the acousmatic interpretation, i.e. the gesture of the sound that is created in a moment, its presence within the room and its propagation from there. "Restored to Life" is a composition for wave field synthesis, where I for the first time try to develop these ideas within an auralisation.'

Ballroom

Markus Schneider + visualisation by *Pietro Sanguineti* - (2003) - 6:00
6 minutes audiovideo for a spatial sound system.
audio: markus schneider
audio source: soundfiles from a webpage that explain the positioning of sound in space. all files are simulating various positions of sound in space. the original files where partially treated and then arranged on 8 tracks within live audio software. the original source now will appear differently and in new spatial position through the spatial sound system .

video: pietro sanguineti.
4 videos on 4 screens show:
video 1 = 1 ball.
video 2 = 2 balls.
video 3 = 3 balls.
video 4 = 4 balls.

this because of 2 reasons:
- each ball is representing the image of an image (here as well an 3d image). there are 4 screens.
- the piece is called ballroom.

Pollocks Sprechwunsch

Marije Baalman - (2003) - 6:02
Jackson Pollocks improvisational drip painting technique was the model for determining the movements of the sounds through space. The word "Sprechwun-

sch" (wish to speak), found in the Berlin trams, was the source of inspiration for the sounds. In this piece balloons try to speak...

Hands and Fingers

Robert Lippok - (2003) - 6:04

Pingpong Ballet

Marc Lingk - (2003) - 15:00
I wanted an impulsive, discontinuous and discretely moving sound, that should not be from a percussion instrument nor an artificial electronic sound. This sound had to be very fast and rich in overtones, so that one can work with its sound color and spaciousness. I then had the choice between pinpong balls and broken beer bottles. The last ones don't fly as fast, and not at all without distortion. So I was left with pingpong balls. That also had an advantage, as the setup of the game itself suggests a choreographie. Left and right two players with bats, a table in the middle, and a ball, a very fast ball, that also from time to time drops on the floor. Modulation algorithms will be played, that could make a real sportsman crazy. The system of Marije Baalman is very suitable to display this movement in space. I see a grand future in this system.
Marc Lingk, 31.1.3

B.2 Electrofringe 2003

Beurskrach

Marije Baalman + visualisation by *Julius Stahl* - (2003) - 3:30
Exploring the possibilities of an object in space, that emanates slightly different from different points. Playing with distance perspective as in a drawing. Imploding and exploding and coming together and finally falling apart. Marije Baalman

Building on photographic material of the sources of sound, each "voice" is given a visual platform whose representation is determined from the property of the respective sound. Therein is a background platform, interpreting the acoustic environment in a different way. The representation varies, with every performance. Julius Stahl

B.3 Linux Audio Conference 2004

Scratch

Marije Baalman
This is a sound installation utilising Wave Field Synthesis, created for the Linux Audio Conference 2004, in Karlsruhe, Germany. Scratch is a sonic creature, that makes scratchy sounds and moves around in space. The creature evolves itself and reacts to some extent on the audience.

B.4 Donaueschingen 2006

Hallenfelder

Kirsten Reese
Klang- Videoinstallation (2006), 50 Minuten
Die Klang-Video-Installation, entstanden 2006 im Auftrag der Donaueschinger Musiktage, wurde für ein Wellenfeldsynthesesystem mit 20 Lautsprechern entwickelt. Der Atmosphäre der leeren Turn- und Mehrzweckhallen Donaueschingens werden auditive und visuelle Momentaufnahmen der vielfältigen Veranstaltungen, die in diesen Räumen das Jahr über stattfinden, gegenübergestellt. "Hallenfelder" handelt von neun Hallen und Sälen, in denen während der Donaueschinger Musiktage regelmäßig Musikveranstaltungen stattfinden. Einerseits werden Klang und Atmosphäre der Hallen selbst in den Mittelpunkt gestellt. Andererseits geht es um das Potenzial dieser Räume - als "Mehrzweckhallen" zeichnen sie sich dadurch aus, dass sie mit unterschiedlichen Inhalten gefüllt werden können. In die Stille der leeren Räume brechen auditive und visuelle Momentaufnahmen von Veranstaltungen ein, die dort das Jahr über stattfinden, wenn nicht Musiktage sind. Durch mediale Aufzeichnung und Bearbeitung werden reale Klänge abstrahiert und "musikalisiert". Gleiches geschieht mit den Eindrücken der Veranstaltungen selbst: was erklingt, ist beispielhaft, aber auch beliebig, ist singulär, aber auch immer wieder gleich. Die Ereignisse sind - wie die Musiktage selbst - einmalig und wiederholen sich immer wieder, je nach Perspektive und Bedeutung, die ihnen auf einer persönlichen, kulturellen und künstlerischen Ebene zugewiesen werden. "Hallenfelder" wird über eine Lautsprecherzeile mit 20 Lautsprechern in Wellenfeldsynthese wiedergegeben. Da es nicht um die Abbildung von Realität, sondern eher um ihre Zuspitzung geht, werden die Möglichkeiten der Wellenfeldsynthese nicht nur für eine Verdeutlichung der räumlichen Klangperspektiven genutzt, sondern auch um mit "unnatürlich" wechselnden Hörperspektiven, momenthaften Irritationen und Illusionen zu spielen.

B.5 Linux Audio Conference 2007

WFSLoop

Wave Field Synthesis (WFS) is an interesting method for spatialisation of electronic music. Its main advantage is that it has no sweet spot, but instead a large listening area, making the technology attractive for concert situations.

The concept of Wave Field Synthesis (WFS) is based on a principle that was thought of in the 17th century by the Dutch physicist Huygens (1690) about the propagation of waves. He stated that when you have a wavefront, you can synthesize the next wavefront by imagining on the wavefront an infinite number of small sources, whose waves will together form the next wavefront.

Based on this principle, Berkhout (1988) introduced the wave field synthesis principle in acoustics.

By using a discrete, linear array of loudspeakers, one can synthesize correct wavefronts in the horizontal plane. An interesting feature is that it is also possible to synthesize a sound source in front of the speakers, something which is not possible with other techniques.

There are some limitations to the technique.

Spatial aliasing has a result that a wave field is not correctly synthesized anymore and artefacts occur. This results in a bad localisable sound source. This limitation is a physical limitation, which can not really be overcome. However it depends on the sound material whether or not this aliasing is a problem from a listener's point of view. In general, if the sound contains a broad spectrum with enough frequencies below the aliasing frequency, the source is still well localisable.

Furthermore, a lot of loudspeakers are needed as well as computation power to calculate the speaker signals for all these speakers.

For the Wave Field Synthesis system in lecture hall H0104, four compositions have been prepared, which will play in a loop in the times mentioned below. The loop starts every hour.

Reale Existenz! (2007)

André Bartetzki
This piece is based on short fragments of a lecture by the Austrian physicist Schrödinger. Erwin Schrödinger, one of the inventors of quantum physics, got very popular due to his thought experiment with a cat in a closed box in which he tried to illustrate the superposition of quantum states. Coupled to the state of a decaying atom (via a Geiger counter and a flask of acid) the cat is after a while both dead and still alive according to the superposition of the two possible states of that unstable atom. Only a collapse of the wave function of this system - caused by an observer or by the influence of the macroscopic environment - could the cat release of its indecisive state.

Streams (2007)

Victor Lazzarini
Streams is a multi-dimensional woodwind quartet, set in 3-D physical space and in the several dimensions of frequency-space. The piece is roughly composed of three overlapping sections, starting with a slow continuous-sound part where each of the four instruments get split in two and glide around the space eventually condensing back together. The middle section starts with chord-forming lines that are spun around the space, continuing into brief statements of melodic motives that eventually lead to interlocking ostinatos. These accellerate to an impossible tempo and then slow down to the original speed. The third section is dominated by an ever-ascending canon (by tone), played by slightly un-synchronised parts. A final coda is based on the ideas/texture of the first section. The piece was composed mostly using software specially developed by the composer for spectral and time-domain processing. This wavefield-synthesis version is dedicated to the TU-Berlin WFS team: Marije, Simon, Torben, Thilo, Daniel and Eddie.

Rituale

Hans Tutschku - (2004) - 15:00

The 15 minute piece "Rituale" (2004) processes human voices and instrumental sounds from various cultures to a sound ritual. It is a continuation of the work on "Rojo" and "object-obstacle", which were both also concerned with the theme of rituals. The composition uses extensively the possibility to place sound sources between the speakers and the listeners - and so inside the listening room. In this way the sounds come very close to the listeners.

"Rituale" was originally (in 2004) created for the IOSONO Wave Field Synthesis system in the Ilmenauer Lindenkino, and was adapted for the lecture hall of the TU Berlin in 2007.

East (from *Atlas*)

Christian Calon - (2007) - 17:00

Based on an idea of cartography, the Atlas project, in its final form will be a *spatial installation*. Musically, it is an hommage to man's creative enterprise which consists into probing the unknown with the help of sound making instruments and then to turn ephemeral impressions into imperishable creations.

This part of Atlas, *East*, was realized first as a concert version. The composition and the spatialization followed the concept of soundfield, which a WFS excels at reproducing. All sounds were generated as correlated multitrack objects to be placed and spatialized as small constellations in this larger sound field, in order to create a multi-directional space for the listener.

The music is based on the sounds of traditional music instruments from the Far Eastern regions of the Earth.

The realization of East was made possible with a commission from the Inventionen Festival/DAAD, Berlin, and the Canada Council. I am very grateful to these institutions whom I warmly thank for triggering and supporting this stage of the large Atlas project.

In its final spatial installation form, Atlas will stage several parallel maps:

- in sound, it is a tribute to the creative enterprises of man in probing the unknown with the help of his musical instruments
- as a silent and virtual monument, the projected *bodyscape* images will present as another cartography, a surround topography of skins and faces of mankind in its richness and diversity
- at the same time projected texts appearing on or around the bodyscapes, will draw and list an underlying world map of *Infamies*, acts of hate, violence and power, perpetrated by man on his fellow man.

B.6 Lange Nacht der Wissenschaften 2007

Xronos

Ludger Brümmer - 18.10 min.sek

Elektroakustische Komposition und Video-Tryptichon

Marije Baalman - On wave field synthesis and electro-acoustic music, ...

Tanz: *Nick Haffner, Katja Büchtermann*
Video, Computer Animationen, Musik: *Ludger Brümmer*

Zeit, Ewigkeit, verändern, entwickeln, vergessen, Sein; es gibt viele Sichtweisen auf die Zeit, seien es statische, dynamische oder philosophische. Zeit ist die Grundlage für alles prozesshaft und läßt sich nur mit sich selbst, über den Weg der Geschwindigkeit verändern. Die Musik, als akustische Äußerung der Zeit benutzt allesamt mit physikalischen Modellen erzeugte Klänge. Diese sind zu mehr oder weniger prozeßhaften Organismen zusammengefasst und bilden eine große formale Phrase, die sich von einem weniger dichten Prozeß in einen repititiven Sog hinein steigert. Anders als die Musik, benutzt die visuelle Ebene neben den Computeranimationen auch dynamische Phänomene der Natur, wie Wasser und Wolken, aber auch Individuen, Tänzer. Alle diese Objekte bewegen sich. Jedes auf seine eigenen Art, die sich jedoch auch untereinander in Beziehung setzen lassen, ob sie nun Gestalten oder Gesten ausbilden oder ob ihre Bewegungen in einer Fläche untergehen. Der Film soll weder die Musik erklären, noch soll er sie kommentieren. Er soll genauso wie der Klang und der Rhythmus eine Ausdrucksweise der Zeit darstellen.

Der Zuspielung über die WFS-Analge stehen 3 Videos gegenüber. Die Videos selbst enthalten zwei oder einen sich bewegenden Körper, die Animationen enthalten 5 einzelne Elemente. Die dreifache Aufteilung wird mal zum Gesamtbild, mal zur Einzelgestalt. Jegliche Kombination ist denkbar und es ist wichtig die Dinge auseinanderzuhalten, nicht zusammenzuführen. In manchen Momenten jedoch addieren sich alle Elemente zusammen. Xronos entstand anlässlich eines Kompositionsauftrages des französischen Kultusministeriums und wurde mit der Software Genesis von ACROE Grenoble entwickelt. Es wurde am Sonic Arts Reserch Center in Belfast und ZKM in Karlsruhe produziert.

Oral 29

Michael Ammann
"Oral 29 beschreibt meine Forschung und mein Fortschreiten innerhalb meiner persönlichen, akustischen Impression und einer "abstrakten Musikalität". Die einzelnen Kompositionen sind als nie fertige Skizzen und prozesshaft-intendiert zu verstehen. Sämtliche Klänge sind phonetisch generiert, sprich meine Stimme dient als einzige Klangquelle (siehe oralPhonetik). Mittels meinem Surround Tools `Q_TIP` bin ich in der Lage Klänge räumlich zu verorten. Ohne softwaregestütze Bearbeitungsund-Automatisierungsprozesse und ohne digitales Filtering ist die Arbeit in dieser Form nicht möglich."

Zellen - Linien

Hans Tutschku - (2007)
(ca. 13 Minuten), Uraufführung
für Klavier und live-Elektronik, mit *Sebastian Berweck*

Die elektroakustischen Bearbeitungen der Klavierklänge in Echtzeit werden alle durch die Spielweise des Pianisten gesteuert. Während des Spiels werden die musikalischen Gesten des Instrumentalisten analysiert und in Abhängigkeit derer vorbestimmte Reaktionen im elektroakustischen Part ausgelöst. Dies bietet dem Spieler eine sehr direkte Kommunikation mit seinem “Begleiter”. Die Energie des Spielers erfährt eine direkte Entsprechung in den Klangbearbeitungen und derer Verräumlichung. Die Wiedergabe der Elektronik mit dem System der Wellenfeldsynthese erlaubt eine genaue Abbildung von Klangbewegungen, die im Dialog zum Klavierpart stehen.

Immersive Motion Study

Shintaro Imai - (2007)
This electroacoustic piece was composed for the Wave Field Synthesis System at TU Berlin, and was realized at Kunitachi College of Music in Tokyo and Electronic Music Studio TU Berlin. The concept of this piece is to experiment on relations between spatial movement/position/ direction of electroacoustics and transition of its timbre/texture. Such, "motion of sound" as it were, represent immersive perspectives in musical dynamics. The sound materials were processed and organized via a realtime algorithmic sound-generating system based on various granular sampling and phase vocoding technique. Some of sound sources were performed by flutist Sabine Vogel. The compositional algorithm and sound synthesis programs were all written in Max/MSP. For editing and mastering, Pro Tools was used as well.

Appendix C

Algorithms

Algorithm C.1 Numbering planes in the mesh

```
function WFSMesh::number planes
    for all triangles do
        if triangle has no plane number then
            create new plane
            for all triangles do
                if in the same plane then
                    assign plane number
                end if
            end for
        end if
    end for
end function
```

Algorithm C.2 Finding planes and outer edges in the mesh

```
function WFSMesh::find planes and outeredges( npoints )
    if not created planes and outeredges then
        number planes
        for all triangles T do
            get the associated plane and add the triangle T to it
            find all the neighbouring triangles
            find all the neighbouring triangles on the same plane
            if more neighbours than on the same plane then
                                ▷ this means the triangle is near an outeredge
                                ▷ find the edges that are part of the outeredge
                for all edges E of T do
                    for all triangles T2 the edge is part of do
                        if not on same plane as T then
                            store reference to this E
                        end if
                    end for
                end for
                for all found edges of T do
                    for all outeredges do
                        if edge part of outeredge then
                            not a new edge
                            keep reference to this outeredge
                        end if
                    end for
                    if new edge then
                        create new outeredge                    ▷ see text
                        create midpoints for outeredge
                    end if
                    add outeredge to plane
                end for
            end if
        end for
    else if npoints != created npoints then
        Redefine the midpoints on all outeredges
        if diffraction calculated then
            Reset the edgesources
        end if
    end if
end function
```

Algorithm C.3 3D WFS point source calculation

```
function WFSOperator::calcpointfilt( WFSMesh, Array)
    for all vertices do
        if filter != zero then
            create WFSPoint
            for all ArraySpeakers do
                calc point delay att( speaker, vertex, mesh )
            end for
        end if
    end for
end function

function WFSOperator::calc point delay att(ArraySpeaker, Vertex, WFSMesh)
    calculate stationary point
    calculate distance to stationary point
    calculate distance between reference line and stationary point
    if point visible from speaker then
        if point behind speaker then
            calculate delay
            calculate attenuation
            point valid
        else                                        ▷ not implemented yet
            calculate delay
            calculate attenuation
            point valid
        end if
    else
        delay and attenuation zero
        point not valid
    end if
end function
```

Algorithm C.4 Speaker response calculation per object

```
function OBJECT::CALCULATEANDSAVEWFS( Array, filename)
    if calculate diffraction then
        mesh->find planes and outer edges
    end if
    mesh->estimate filter length
    mesh->set windowsize
    mesh->set samplerate
    if point source then
        create mesh with 1 point
    end if
    mesh->scale
    mesh->create filter impulse responses
    if calculate diffraction then
        mesh->create diffraction edges                          ▷ see section 6.6.
        mesh->prepare diffraction irs
    end if
    for all Location do
        mesh->transform to location
        if calculate diffraction then
            mesh->create diffraction edgelines   ▷ this ensures the coordinates
of the edgelines are correct for this location
        end if
        if source type == monopole then
            wfsoperator->calculate point source filters( array, mesh )
        end if
        if source type == dipole then
            wfsoperator->calculate dipole source filters( array, mesh )
        end if
        if calculate diffraction then
            wfsoperator->calculate diffraction point filters( array, mesh )
        end if
        wfsoperator->init convolution( mesh )
        WFSOPERATOR->CALCULATEANDSAVEFILTERS                ▷ see alg. C.5
        mesh->transform back
    end for
    mesh->undo scaling
end function
```

Algorithm C.5 Speaker response calculation per location

```
function WFSOperator::calculateAndSaveWFS( filename, WFS-
Mesh, ObjectCoordinates )
  for all ArraySpeakers do
    initialise speakerfilter
    for all WFSPoints do
      if attenuation not zero then
        prepare FilterNode filter for convolution
        if filter == unity then
          put delay and attenuation in speaker filter
        else
          create a WFS filter based on delay and attenuation
          convolve with the FilterNode filter
          put result in speaker filter
        end if
      end if
    end for
    if calculate diffraction then                     ▷ see section 6.6.
      for all DiffractionEdges do
        if valid then
          calculate impulse response
          if impulse response non zero then
            put result in speaker filter
          end if
        end if
      end for
    end if
    write CoefSet information
    multiply speaker filter with amplitude factor     ▷ see text
    save speaker filter to file
  end for
end function
```

Algorithm C.6 The algorithm to determine the diffraction edges

```
function WFSMesh::create diffraction edges ( order, min. distance, tolerance, z_range )
  if created diffraction order < order then
                                   ▷ create the first order diffraction edges
    if created diffraction order == 0 then
      for all Planes P do
        for all OuterEdges O in P do
          counter diffraction edges for this edge n = 0
          for all Vertices in P do
            if Filter ! = zero then
              if Vertex not on O then
                if n == 0 then
                  create new diffraction edge with vertex as source
                else
                  add vertex as a source to diffraction edge
                end if
              end if
            end if
          end for
        end for
      end for
    end if
                                   ▷ create the higher order diffraction edges
    for max( created diffraction , 2 ) up to order do
      for all OuterEdges O do
        counter diffraction edges for this edge n = 0
        for all Planes P, O is part of do
          for all OuterEdges O2 of P do
            if O! = O2 then
              get list of diffraction edges in O2 that are of a lower order than the one we are currently creating
              for all diffraction edges D in this list do
                if n == 0 then
                  create new diffraction edge Dn
                  with D as lower edge
                else
                  add Dn as a higher edge to D
                end if
              end for
            end if
          end for
        end for
      end for
    end for
  end if
  created diffraction = order
end function
```

Algorithm C.7 The algorithm to calculate the intermediary impulse responses of the diffraction edges

```
function WFSMesh::prepare diffraction irs
    for order = 1 to highest order - 1 do
        for all diffraction edges of order do
            calculate the impulse response
        end for
    end for
end function

function DiffractionEdge::calculate irs
    if order == 1 then                                          ▷ first order
        for all higher edges H do
            for all diffraction sources D on H do
                create an IR vector for the result and initialise at zero
                set D as the receiver for this edge
                for all diffracted sources S do
                    set S as source for this edge
                    calculate the edge source response      ▷ see algorithm C.8
                    add result to IR
                end for
                if IR is zero then
                    delete IR
                else
                    store IR in D
                end if
            end for
        end for
    else if order > 1 then                                      ▷ higher orders
        for all higher edges H do
            for all diffraction sources D on H do
                create an IR vector for the result and initialise at zero
                set D as the receiver for this edge
                for all lower edges L of this edge do
                    for all diffraction sources S of L do
                        set S as source for this edge
                        calculate the edge source response  ▷ see algorithm C.8
                        add result to IR
                    end for
                end for
                if IR is zero then
                    delete IR
                else
                    store IR in D
                end if
            end for
        end for
    end if. with a focus on the reproduction of arbitrarily shaped sound sources
end function
```

Algorithm C.8 Algorithm for calculating the response for a diffraction edge, with set source and receiver

function DIFFRACTIONEDGE::CALCULATE EDGESOURCE RESPONSE
initialise result IR_R
find sample aligned sources
find within which edge sources the sample aligned are
for all sample aligned diffraction sources S on this edge **do**
calculate the contribution Δh_i for S
add contribution at sample time to IR_R (first order only)
end for
for all diffraction sources D on this edge **do**
calculate the contribution Δh_i for D
end for
for all diffraction sources D on this edge **do**
calculate response IR_D for D with (first order only) slope correction
if order > 1 **then**
convolve with IR from previous order
end if
add IR_D to IR_R
end for
end function

function DIFFRACTIONSOURCE::CALCULATE IR
Calculate distance of edge segment end points to receiver lc_1, lc_2
Calculate the corner samples
if no sample aligned segments within this segment **then**
calculate start and end delay
do numerical integration (calculate contribution quad)
else if not only sample aligned segments within this segment **then**
for all each end that does not belong to sample aligned segment **do**
calculate the length in samples
calculate a midpoint z_{mid}
do numerical integration (calculate contribution quad)
calculate a midsample (needed for slope correction)
end for
end if
end function

function SAMPLEALIGNEDSOURCE::CALCULATE IR
if not an apex segment **then**
do numerical integration (calculate contribution quad)
else if apex segment **then**
calculate apex contribution for part within z_{range} ▷ within z_{range}
calculate midpoint for z_{range}-z_{near} to ▷ z_{range} to nearest endpoint
do numerical integration (calculate contribution quad)
calculate midpoint for z_{range}-z_{far} ▷ z_{range} to furthest endpoint
do numerical integration (calculate contribution quad)
add all three contributions
end if
end function

Appendix D

Listening test instructions

Hörtest zur Wiedergabe von willkürlich geformten Sound-Quellen mit Hilfe der Wellenfeldsynthese - Teil 1

Vielen Dank für die Teilnahme an diesem Hörtest!

Hintergrund

In herkömmlichen Wellenfeldsynthese Anwendungen ist nur die Wiedergabe von Punktquellen oder Ebenen Wellen möglich. Neue Forschungen untersuchen ob und wie willkürlich geformte Schall-quellen widergegeben werden können.

Das Ziel dieses Hörtestes ist es, herauszufinden welche Effekte bei der Reproduktion einer willkürlich geformten Schallquelle hörbar sind.

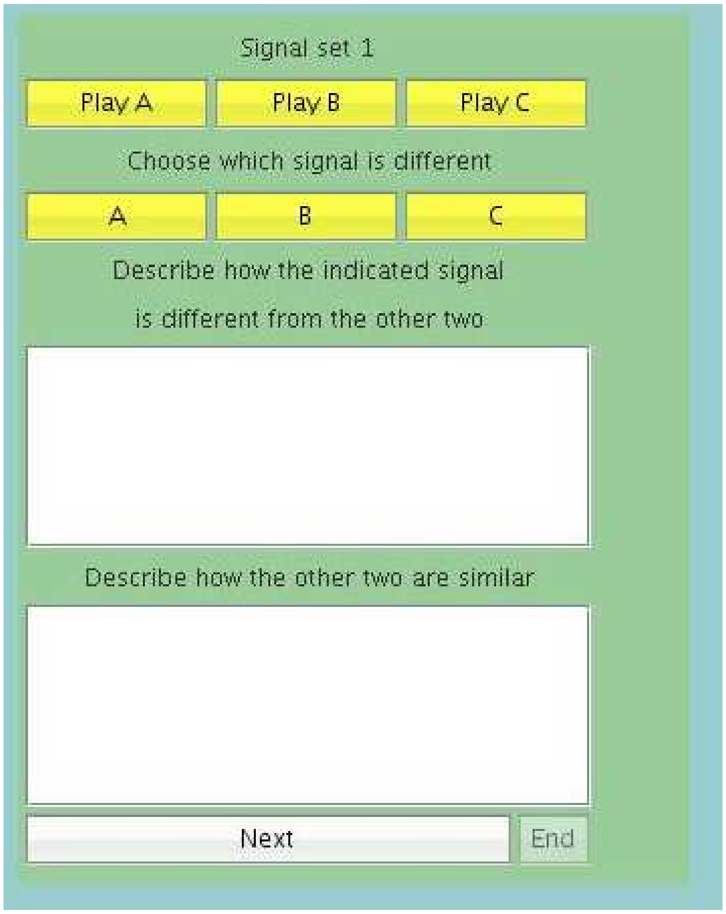
Evaluierung

Dir werden in insgesamt 27 Durchgängen jeweils 3 verschiedene Signale vorgespielt. Jedes dieser Signale ist maximal 10 Sekunden lang. Du kannst dir in jedem Durchgang die Signale so oft anhören wie du möchtest.

Aufgabe:

In jedem Durchgang sollst du die drei vorgespielten Signale vergleichen. Gib an, welches Signal sich im Höreindruck am meisten gegenüber den beiden anderen Signalen unterscheidet. Beschreibe bitte diesen Unterschied. Bitte versuche die Unterschiede so detailiert und genau wie möglich zu beschreiben.
Erkläre zusätzlich, was die anderen beiden Signale gemeinsam haben im Höreindruck. Gib hier bitte auch eine möglichst genaue Beschreibung.
Die Beurteilung der wahrgenommenen Unterschiede und Gemeinsamkeiten soll

auf akustisch wahrnehmbare Eigenschaften der Signale basieren und *nicht* auf deiner persönlichen Präferenz (was du magst oder nicht magst).



Hörtest zur Wiedergabe von willkürlich geformten Sound-Quellen mit Hilfe der Wellenfeldsynthese - Teil 2

Vielen Dank für die Teilnahme an diesem Hörtest!

Hintergrund

In herkömmlichen Wellenfeldsynthese Anwendungen ist nur die Wiedergabe von Punktquellen oder Ebenen Wellen möglich. Neue Forschungen untersuchen ob und wie willkürlich geformte Schallquellen widergegeben werden können.

Das Ziel dieses Hörtestes ist es, herauszufinden welche Effekte bei der Reproduktion einer willkürlich geformten Schallquelle hörbar sind.

Evaluation phase

Dir werden in insgesamt 12 Durchgängen jeweils 4 oder 5 verschiedene Signale vorgespielt. Jedes dieser Signale ist maximal 20 Sekunden lang. Du kannst dir in jedem Durchgang die Signale so oft anhören wie du möchtest.

Aufgabe:

In jedem Durchgang sollst du die wiedergegebenen Signale nach bestimmten Merkmalen bewerten. Diese Merkmale wurden im vorhergehenden Experiment ermittelt. Zu beachten ist, dass Die Beurteilung der wahrgenommenen Unterschiede und Gemeinsamkeiten auf akustisch wahrnehmbaren Eigenschaften der Signale basieren soll und nicht auf deiner persönlichen Präferenz (was du magst oder nicht)

Skala:

Jedes Merkmal ist durch eine bipolaren Skala gekennzeichnet. Diese Skala ist in 5 Stufen unterteilt und hat zwei entgegenesetzte Ausprägungen als Maxima. Ein Beispiel für die Einteilung der Skala kannst du nachfolgend erkennen.

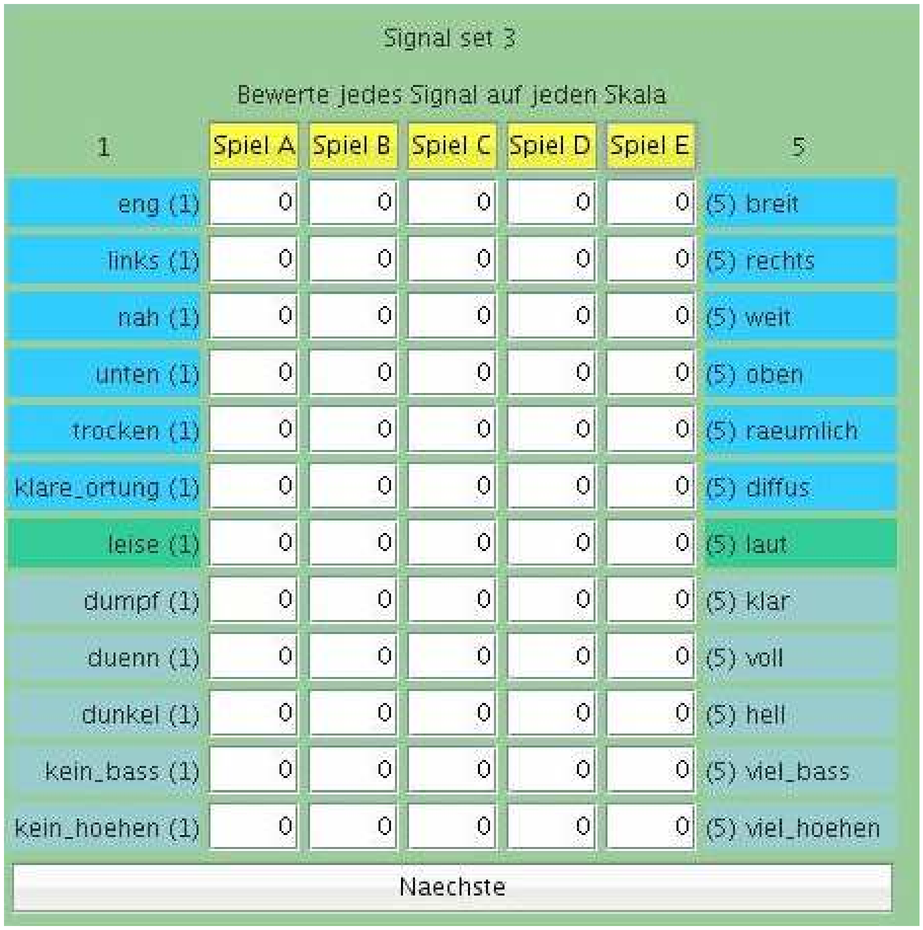
1	2	3	4	5
sehr groß	groß	mittel	klein	sehr klein

In diesem Hörtest sollst du jedem Signal für jedes Merkmal einen Wert von der 5-stufigen Skala zuordnen. Dabei ist in jeder Zeile das jeweilige Merkmal mit seinen zwei bipolaren Ausprägungen aufgetragen. In jeder Spalte mußt du den Bewertungswert der Skala für das entsprechende Signal eintragen. Im nachfolgenden Schema ist beispielhaft die Bewertung von vier Signalen und zwei Merkmalen dargestellt.

	Signal 1	Signal 2	Signal 3	Signal 4	
sehr groß	4	4	3	3	sehr klein
realistisch	3	1	4	2	unrealistisch

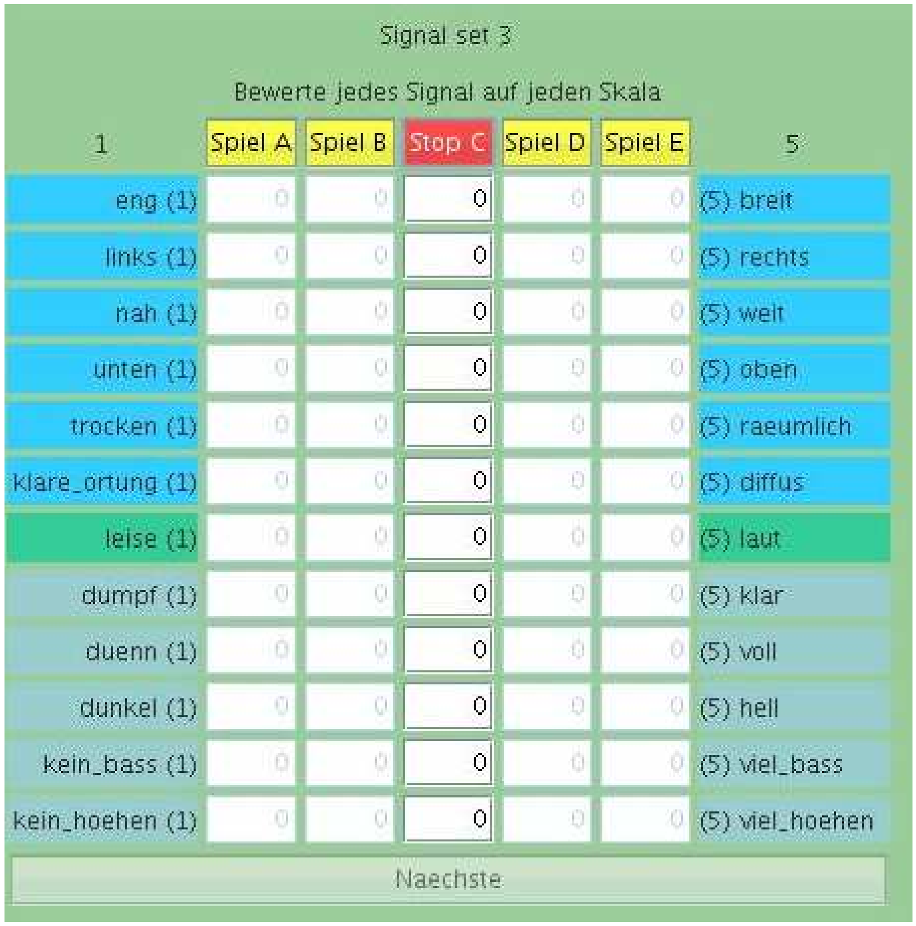
Hörtestoberfläche:

Die Benutzeroberfläche sieht so aus:



Wenn du ein Signal abspielst, werden die Spalten der andere Signale ausgeblendet und kannst du das Signal welches spielt bewerten auf den Skalen. Du kannst die Zahlen 1 bis 5 eintragen, oder mit den Pfeilen-Tasten (oben, unten) die Werte ändern. Mit <tab> gehst du zum nächsten Merkmal.

Die ersten 6 Merkmale haben Bezug auf Räumlichkeit, das 7. Merkmal auf Lautheit, und die letzten 5 auf Klangfarbe.


Signal set 3
Bewerte jedes Signal auf jeden Skala
1
Spiel A
Spiel B
Stop C
Spiel D
Spiel E
5
eng (1)
(5) breit
links (1)
(5) rechts
nah (1)
(5) weit
unten (1)
(5) oben
trocken (1)
(5) raeumlich
klare_ortung (1)
(5) diffus
leise (1)
(5) laut
dumpf (1)
(5) klar
duenn (1)
(5) voll
dunkel (1)
(5) hell
kein_bass (1)
(5) viel_bass
kein_hoehen (1)
(5) viel_hoehen
Naechste

Appendix E

Detailed results of the listening tests

Table E.1: Detailed ANOVA results for the Refinement set

Refinement	Sum Sq	Df	F value	Pr($> F$)
Refinement: eng - breit				
fdistance	2.786	1.000	2.349	0.126
fposition	6.842	1.000	5.770	0.017
signal	42.789	3.000	12.027	0.000
stimulus	0.091	1.000	0.077	0.782
fdistance:fposition	0.488	1.000	0.412	0.521
fdistance:signal	6.810	3.000	1.914	0.126
fposition:signal	0.332	3.000	0.093	0.964
fdistance:stimulus	0.372	1.000	0.314	0.575
fposition:stimulus	0.568	1.000	0.479	0.489
signal:stimulus	1.308	3.000	0.368	0.776
fdistance:fposition:signal	0.528	3.000	0.148	0.931
fdistance:fposition:stimulus	0.034	1.000	0.029	0.865
fdistance:signal:stimulus	6.107	3.000	1.717	0.163
fposition:signal:stimulus	1.815	3.000	0.510	0.675
fdistance:fposition:signal:stimulus	1.909	3.000	0.537	0.657
Residuals	569.219	480.000		
Refinement: links - rechts				
fdistance	5.544	1.000	8.649	0.003
fposition	8.787	1.000	13.711	0.000
signal	15.227	3.000	7.919	0.000

continued on next page

continued from previous page				
Refinement	Sum Sq	Df	F value	Pr(> F)
stimulus	1.371	1.000	2.139	0.144
fdistance:fposition	0.837	1.000	1.306	0.254
fdistance:signal	7.337	3.000	3.816	0.010
fposition:signal	5.297	3.000	2.755	0.042
fdistance:stimulus	0.055	1.000	0.085	0.771
fposition:stimulus	0.537	1.000	0.838	0.360
signal:stimulus	0.544	3.000	0.283	0.838
fdistance:fposition:signal	33.836	3.000	17.597	0.000
fdistance:fposition:stimulus	3.018	1.000	4.708	0.031
fdistance:signal:stimulus	0.847	3.000	0.441	0.724
fposition:signal:stimulus	1.152	3.000	0.599	0.616
fdistance:fposition:signal:stimulus	0.885	3.000	0.460	0.710
Residuals	307.646	480.000		
Refinement: nah - weit				
fdistance	0.504	1.000	0.660	0.417
fposition	5.979	1.000	7.824	0.005
signal	7.734	3.000	3.374	0.018
stimulus	0.577	1.000	0.755	0.385
fdistance:fposition	2.537	1.000	3.320	0.069
fdistance:signal	32.402	3.000	14.134	0.000
fposition:signal	2.841	3.000	1.239	0.295
fdistance:stimulus	6.551	1.000	8.573	0.004
fposition:stimulus	0.068	1.000	0.089	0.765
signal:stimulus	3.020	3.000	1.318	0.268
fdistance:fposition:signal	1.947	3.000	0.849	0.467
fdistance:fposition:stimulus	0.008	1.000	0.011	0.917
fdistance:signal:stimulus	0.884	3.000	0.386	0.763
fposition:signal:stimulus	0.914	3.000	0.399	0.754
fdistance:fposition:signal:stimulus	0.363	3.000	0.158	0.924
Residuals	366.795	480.000		
Refinement: unten - oben				
fdistance	0.145	1.000	0.305	0.581
fposition	3.065	1.000	6.420	0.012
signal	5.195	3.000	3.627	0.013
stimulus	0.500	1.000	1.047	0.307
fdistance:fposition	0.171	1.000	0.358	0.550
fdistance:signal	0.018	3.000	0.012	0.998
fposition:signal	0.114	3.000	0.079	0.971
fdistance:stimulus	0.418	1.000	0.876	0.350
fposition:stimulus	0.926	1.000	1.940	0.164
				continued on next page

continued from previous page

Refinement	Sum Sq	Df	F value	Pr(> F)
signal:stimulus	0.233	3.000	0.162	0.922
fdistance:fposition:signal	2.109	3.000	1.472	0.221
fdistance:fposition:stimulus	0.042	1.000	0.088	0.767
fdistance:signal:stimulus	0.272	3.000	0.190	0.903
fposition:signal:stimulus	0.963	3.000	0.673	0.569
fdistance:fposition:signal:stimulus	0.403	3.000	0.281	0.839
Residuals	229.171	480.000		
Refinement: trocken - räumlich				
fdistance	0.003	1.000	0.003	0.959
fposition	13.305	1.000	10.554	0.001
signal	37.521	3.000	9.921	0.000
stimulus	16.931	1.000	13.430	0.000
fdistance:fposition	0.105	1.000	0.084	0.773
fdistance:signal	6.369	3.000	1.684	0.170
fposition:signal	2.270	3.000	0.600	0.615
fdistance:stimulus	1.459	1.000	1.158	0.283
fposition:stimulus	0.261	1.000	0.207	0.649
signal:stimulus	0.851	3.000	0.225	0.879
fdistance:fposition:signal	0.955	3.000	0.252	0.860
fdistance:fposition:stimulus	0.087	1.000	0.069	0.793
fdistance:signal:stimulus	1.149	3.000	0.304	0.823
fposition:signal:stimulus	1.174	3.000	0.311	0.818
fdistance:fposition:signal:stimulus	2.450	3.000	0.648	0.585
Residuals	605.141	480.000		
Refinement: klare Ortung - diffus				
fdistance	5.274	1.000	4.771	0.029
fposition	1.419	1.000	1.284	0.258
signal	233.133	3.000	70.301	0.000
stimulus	0.833	1.000	0.754	0.386
fdistance:fposition	0.195	1.000	0.177	0.674
fdistance:signal	7.640	3.000	2.304	0.076
fposition:signal	0.270	3.000	0.081	0.970
fdistance:stimulus	0.382	1.000	0.346	0.557
fposition:stimulus	0.163	1.000	0.147	0.702
signal:stimulus	2.983	3.000	0.900	0.441
fdistance:fposition:signal	1.675	3.000	0.505	0.679
fdistance:fposition:stimulus	1.534	1.000	1.387	0.239
fdistance:signal:stimulus	2.335	3.000	0.704	0.550
fposition:signal:stimulus	3.606	3.000	1.087	0.354
fdistance:fposition:signal:stimulus	0.687	3.000	0.207	0.891

continued on next page

continued from previous page				
Refinement	Sum Sq	Df	F value	Pr($> F$)
Residuals	530.593	480.000		
Refinement: dumpf - klar				
fdistance	1.290	1.000	2.387	0.123
fposition	2.011	1.000	3.720	0.054
signal	691.648	3.000	426.580	0.000
stimulus	1.972	1.000	3.648	0.057
fdistance:fposition	0.296	1.000	0.548	0.460
fdistance:signal	0.222	3.000	0.137	0.938
fposition:signal	4.222	3.000	2.604	0.051
fdistance:stimulus	0.806	1.000	1.491	0.223
fposition:stimulus	0.640	1.000	1.184	0.277
signal:stimulus	4.258	3.000	2.626	0.050
fdistance:fposition:signal	0.473	3.000	0.292	0.831
fdistance:fposition:stimulus	0.078	1.000	0.144	0.704
fdistance:signal:stimulus	0.854	3.000	0.527	0.664
fposition:signal:stimulus	2.163	3.000	1.334	0.263
fdistance:fposition:signal:stimulus	0.624	3.000	0.385	0.764
Residuals	259.421	480.000		
Refinement: dünn - voll				
fdistance	0.001	1.000	0.001	0.978
fposition	1.271	1.000	1.301	0.255
signal	257.484	3.000	87.895	0.000
stimulus	0.005	1.000	0.005	0.942
fdistance:fposition	1.003	1.000	1.028	0.311
fdistance:signal	0.912	3.000	0.311	0.817
fposition:signal	2.666	3.000	0.910	0.436
fdistance:stimulus	1.836	1.000	1.881	0.171
fposition:stimulus	0.153	1.000	0.156	0.693
signal:stimulus	0.591	3.000	0.202	0.895
fdistance:fposition:signal	0.221	3.000	0.076	0.973
fdistance:fposition:stimulus	0.314	1.000	0.322	0.571
fdistance:signal:stimulus	0.385	3.000	0.132	0.941
fposition:signal:stimulus	2.793	3.000	0.954	0.415
fdistance:fposition:signal:stimulus	0.552	3.000	0.188	0.904
Residuals	468.715	480.000		
Refinement: dunkel - hell				
fdistance	0.440	1.000	0.977	0.323
fposition	0.312	1.000	0.694	0.405
signal	589.021	3.000	436.044	0.000
stimulus	0.014	1.000	0.031	0.861
				continued on next page

continued from previous page

Refinement	Sum Sq	Df	F value	Pr(> F)
fdistance:fposition	0.095	1.000	0.211	0.646
fdistance:signal	0.423	3.000	0.313	0.816
fposition:signal	1.204	3.000	0.891	0.446
fdistance:stimulus	2.673	1.000	5.937	0.015
fposition:stimulus	0.318	1.000	0.706	0.401
signal:stimulus	0.267	3.000	0.198	0.898
fdistance:fposition:signal	1.293	3.000	0.957	0.413
fdistance:fposition:stimulus	0.002	1.000	0.004	0.947
fdistance:signal:stimulus	0.622	3.000	0.461	0.710
fposition:signal:stimulus	1.112	3.000	0.823	0.481
fdistance:fposition:signal:stimulus	0.536	3.000	0.397	0.755
Residuals	216.133	480.000		
Refinement: kein Bass - viel Bass				
fdistance	1.074	1.000	2.403	0.122
fposition	2.435	1.000	5.450	0.020
signal	569.771	3.000	425.005	0.000
stimulus	0.141	1.000	0.315	0.575
fdistance:fposition	0.175	1.000	0.391	0.532
fdistance:signal	1.553	3.000	1.158	0.325
fposition:signal	2.869	3.000	2.140	0.094
fdistance:stimulus	0.013	1.000	0.028	0.867
fposition:stimulus	0.354	1.000	0.793	0.374
signal:stimulus	1.150	3.000	0.858	0.463
fdistance:fposition:signal	0.277	3.000	0.207	0.892
fdistance:fposition:stimulus	1.077	1.000	2.409	0.121
fdistance:signal:stimulus	0.296	3.000	0.221	0.882
fposition:signal:stimulus	0.481	3.000	0.359	0.783
fdistance:fposition:signal:stimulus	1.129	3.000	0.842	0.471
Residuals	214.500	480.000		
Refinement: kein Höhen - viel Höhen				
fdistance	2.327	1.000	4.737	0.030
fposition	0.399	1.000	0.812	0.368
signal	328.053	3.000	222.582	0.000
stimulus	0.925	1.000	1.882	0.171
fdistance:fposition	0.414	1.000	0.842	0.359
fdistance:signal	0.637	3.000	0.432	0.730
fposition:signal	0.789	3.000	0.535	0.658
fdistance:stimulus	0.218	1.000	0.443	0.506
fposition:stimulus	0.069	1.000	0.140	0.709
signal:stimulus	0.082	3.000	0.056	0.983

continued on next page

continued from previous page				
Refinement	Sum Sq	Df	F value	Pr(> F)
fdistance:fposition:signal	0.533	3.000	0.362	0.781
fdistance:fposition:stimulus	0.673	1.000	1.371	0.242
fdistance:signal:stimulus	1.732	3.000	1.175	0.319
fposition:signal:stimulus	0.380	3.000	0.258	0.856
fdistance:fposition:signal:stimulus	0.868	3.000	0.589	0.622
Residuals	235.816	480.000		
Refinement: leise - laut				
fdistance	1.709	1.000	4.031	0.045
fposition	0.044	1.000	0.104	0.747
signal	19.898	3.000	15.649	0.000
stimulus	1.387	1.000	3.273	0.071
fdistance:fposition	0.263	1.000	0.622	0.431
fdistance:signal	5.449	3.000	4.286	0.005
fposition:signal	3.239	3.000	2.548	0.055
fdistance:stimulus	0.606	1.000	1.429	0.232
fposition:stimulus	1.431	1.000	3.377	0.067
signal:stimulus	0.723	3.000	0.569	0.636
fdistance:fposition:signal	2.175	3.000	1.710	0.164
fdistance:fposition:stimulus	0.114	1.000	0.268	0.605
fdistance:signal:stimulus	0.033	3.000	0.026	0.994
fposition:signal:stimulus	0.332	3.000	0.261	0.854
fdistance:fposition:signal:stimulus	0.453	3.000	0.356	0.785
Residuals	203.450	480.000		

Component	1	2	3	4	5	6	7	8	9	10	11	12
eng - breit	−0.203	−0.366	−0.217	−0.439	0.255	−0.367	0.496	−0.063	−0.363	−0.052	−0.012	−0.046
links - rechts	−0.142	−0.345	0.409	−0.351	−0.565	−0.199	−0.248	0.372	−0.007	0.088	0.054	0.007
nah - weit	−0.087	−0.454	0.439	0.269	0.306	−0.201	−0.315	−0.535	−0.004	−0.008	0.015	0.042
unten - oben	0.029	−0.449	−0.175	0.695	−0.329	0.052	0.326	0.183	−0.173	0.038	−0.047	0.004
trocken - räumlich	−0.144	−0.440	−0.146	−0.264	0.090	0.819	−0.097	−0.055	0.037	0.034	0.032	−0.005
klare Ortung - diffus	−0.330	−0.163	−0.094	0.090	0.332	−0.172	0.091	0.362	0.731	0.146	−0.006	0.114
leise - laut	−0.200	−0.058	−0.654	0.064	−0.015	−0.231	−0.647	0.043	−0.197	0.045	0.098	0.030
dumpf - klar	0.418	−0.107	−0.127	−0.150	−0.061	−0.062	0.011	−0.139	0.048	0.474	−0.305	0.654
dünn - voll	−0.321	0.100	−0.175	−0.051	−0.490	−0.043	0.215	−0.588	0.308	0.160	0.315	0.006
dunkel - hell	0.414	−0.165	−0.130	−0.084	−0.006	−0.101	−0.048	−0.082	0.203	0.399	−0.141	−0.733
kein - viel Bass	−0.427	0.118	0.004	0.016	−0.142	0.033	−0.034	−0.126	−0.010	−0.040	−0.866	−0.115
kein - viel Höhen	0.359	−0.231	−0.204	−0.122	−0.176	−0.109	−0.041	−0.118	0.356	−0.743	−0.146	0.059
standard deviation	4.490	1.724	1.174	0.886	0.795	0.704	0.597	0.553	0.474	0.296	0.172	0.136

correlation	breit	l-r	weit	oben	räumlich	ortung	laut	klar	voll	hell	bass	Höhen
eng-breit	1.000	0.230	0.189	0.071	0.321	0.390	0.257	−0.230	0.234	−0.230	0.269	−0.147
links-rechts	0.230	1.000	0.279	0.079	0.215	0.172	−0.025	−0.202	0.157	−0.188	0.229	−0.098
nah-weit	0.189	0.279	1.000	0.213	0.203	0.209	−0.059	−0.146	−0.035	−0.079	0.082	−0.077
unten-oben	0.071	0.079	0.213	1.000	0.168	0.061	0.082	0.077	−0.033	0.126	−0.112	0.178
trocken-rämlich	0.321	0.215	0.203	0.168	1.000	0.254	0.170	−0.165	0.128	−0.146	0.191	−0.065
klare Ortung-diffus	0.390	0.172	0.209	0.061	0.254	1.000	0.322	−0.576	0.347	−0.492	0.526	−0.421
leise-laut	0.257	−0.025	−0.059	0.082	0.170	0.322	1.000	−0.273	0.304	−0.246	0.361	−0.162
dumpf-klar	−0.230	−0.202	−0.146	0.077	−0.165	−0.576	−0.273	1.000	−0.503	0.852	−0.780	0.700
dünn-voll	0.234	0.157	−0.035	−0.033	0.128	0.347	0.304	−0.503	1.000	−0.530	0.676	−0.396
dunkel-hell	−0.230	−0.188	−0.079	0.126	−0.146	−0.492	−0.246	0.852	−0.530	1.000	−0.798	0.732
kein-viel Bass	0.269	0.229	0.082	−0.112	0.191	0.526	0.361	−0.780	0.676	−0.798	1.000	−0.682
kein-viel Höhen	−0.147	−0.098	−0.077	0.178	−0.065	−0.421	−0.162	0.700	−0.396	0.732	−0.682	1.000

Table E.2: Results of the principal components analysis and the correlation table, for the refinement set

Table E.3: Detailed ANOVA results for the Shape set

Shape	Sum Sq	Df	F value	Pr($> F$)
Shape eng-breit				
f distance	4.883	1.000	4.494	0.035
f position	0.633	1.000	0.582	0.446
signal	153.672	3.000	47.148	0.000
stimulus	0.945	1.000	0.870	0.351
f distance:f position	0.000	1.000	0.000	1.000
f distance:signal	6.820	3.000	2.093	0.100
f position:signal	8.164	3.000	2.505	0.058
f distance:stimulus	2.531	1.000	2.330	0.128
f position:stimulus	0.781	1.000	0.719	0.397
signal:stimulus	3.695	3.000	1.134	0.335
f distance:f position:signal	1.234	3.000	0.379	0.768
f distance:f position:stimulus	0.008	1.000	0.007	0.932
f distance:signal:stimulus	0.672	3.000	0.206	0.892
f position:signal:stimulus	2.328	3.000	0.714	0.544
f distance:f position:signal:stimulus	1.352	3.000	0.415	0.743
Residuals	521.500	480.000		
Shape links-rechts				
f distance	0.158	1.000	0.244	0.622
f position	3.283	1.000	5.062	0.025
signal	44.256	3.000	22.745	0.000
stimulus	0.049	1.000	0.075	0.784
f distance:f position	19.924	1.000	30.720	0.000
f distance:signal	10.209	3.000	5.247	0.001
f position:signal	39.459	3.000	20.280	0.000
f distance:stimulus	1.643	1.000	2.533	0.112
f position:stimulus	0.705	1.000	1.087	0.298
signal:stimulus	2.193	3.000	1.127	0.338
f distance:f position:signal	71.193	3.000	36.590	0.000
f distance:f position:stimulus	0.018	1.000	0.027	0.869
f distance:signal:stimulus	1.037	3.000	0.533	0.660
f position:signal:stimulus	2.037	3.000	1.047	0.371
f distance:f position:signal:stimulus	0.850	3.000	0.437	0.727
Residuals	311.313	480.000		
Shape nah-weit				
f distance	2.258	1.000	3.272	0.071
f position	0.633	1.000	0.917	0.339

continued on next page

continued from previous page

Shape	Sum Sq	Df	F value	Pr($> F$)
signal	40.766	3.000	19.691	0.000
stimulus	0.008	1.000	0.011	0.915
f distance:f position	0.031	1.000	0.045	0.832
f distance:signal	48.445	3.000	23.400	0.000
f position:signal	4.070	3.000	1.966	0.118
f distance:stimulus	0.281	1.000	0.408	0.524
f position:stimulus	0.500	1.000	0.725	0.395
signal:stimulus	1.070	3.000	0.517	0.671
f distance:f position:signal	3.109	3.000	1.502	0.213
f distance:f position:stimulus	0.633	1.000	0.917	0.339
f distance:signal:stimulus	2.172	3.000	1.049	0.371
f position:signal:stimulus	2.078	3.000	1.004	0.391
f distance:f position:signal:stimulus	0.695	3.000	0.336	0.799
Residuals	331.250	480.000		
Shape unten-oben				
f distance	0.002	1.000	0.005	0.945
f position	0.002	1.000	0.005	0.945
signal	7.131	3.000	5.732	0.001
stimulus	0.002	1.000	0.005	0.945
f distance:f position	0.330	1.000	0.796	0.373
f distance:signal	0.584	3.000	0.469	0.704
f position:signal	0.428	3.000	0.344	0.794
f distance:stimulus	0.049	1.000	0.118	0.732
f position:stimulus	0.439	1.000	1.060	0.304
signal:stimulus	0.709	3.000	0.570	0.635
f distance:f position:signal	0.756	3.000	0.608	0.610
f distance:f position:stimulus	0.330	1.000	0.796	0.373
f distance:signal:stimulus	0.756	3.000	0.608	0.610
f position:signal:stimulus	0.021	3.000	0.017	0.997
f distance:f position:signal:stimulus	0.350	3.000	0.281	0.839
Residuals	199.063	480.000		
Shape trocken-räumlich				
f distance	0.500	1.000	0.386	0.535
f position	2.531	1.000	1.956	0.163
signal	98.297	3.000	25.316	0.000
stimulus	7.031	1.000	5.433	0.020
f distance:f position	0.031	1.000	0.024	0.877
f distance:signal	1.609	3.000	0.414	0.743
f position:signal	3.859	3.000	0.994	0.395
f distance:stimulus	0.281	1.000	0.217	0.641

continued on next page

continued from previous page

Shape	Sum Sq	Df	F value	Pr(> F)
f position:stimulus	0.125	1.000	0.097	0.756
signal:stimulus	0.953	3.000	0.245	0.865
f distance:f position:signal	5.859	3.000	1.509	0.211
f distance:f position:stimulus	0.125	1.000	0.097	0.756
f distance:signal:stimulus	2.953	3.000	0.761	0.517
f position:signal:stimulus	4.328	3.000	1.115	0.343
f distance:f position:signal:stimulus	2.141	3.000	0.551	0.647
Residuals	621.250	480.000		
Shape klare Ortung-diffus				
f distance	1.221	1.000	1.281	0.258
f position	1.424	1.000	1.494	0.222
signal	178.287	3.000	62.377	0.000
stimulus	0.049	1.000	0.051	0.821
f distance:f position	0.049	1.000	0.051	0.821
f distance:signal	18.428	3.000	6.447	0.000
f position:signal	3.412	3.000	1.194	0.312
f distance:stimulus	1.033	1.000	1.084	0.298
f position:stimulus	0.096	1.000	0.100	0.751
signal:stimulus	9.693	3.000	3.391	0.018
f distance:f position:signal	3.100	3.000	1.084	0.355
f distance:f position:stimulus	1.033	1.000	1.084	0.298
f distance:signal:stimulus	0.396	3.000	0.139	0.937
f position:signal:stimulus	7.021	3.000	2.457	0.062
f distance:f position:signal:stimulus	2.771	3.000	0.970	0.407
Residuals	457.312	480.000		
Shape dumpf-klar				
f distance	1.643	1.000	2.249	0.134
f position	0.002	1.000	0.003	0.959
signal	326.396	3.000	148.970	0.000
stimulus	1.221	1.000	1.671	0.197
f distance:f position	1.033	1.000	1.415	0.235
f distance:signal	37.053	3.000	16.911	0.000
f position:signal	1.850	3.000	0.844	0.470
f distance:stimulus	2.674	1.000	3.661	0.056
f position:stimulus	1.033	1.000	1.415	0.235
signal:stimulus	2.975	3.000	1.358	0.255
f distance:f position:signal	0.193	3.000	0.088	0.966
f distance:f position:stimulus	0.236	1.000	0.324	0.570
f distance:signal:stimulus	2.959	3.000	1.351	0.257
f position:signal:stimulus	0.381	3.000	0.174	0.914

continued on next page

continued from previous page

Shape	Sum Sq	Df	F value	Pr(> F)
f distance:f position:signal:stimulus	1.428	3.000	0.652	0.582
Residuals	350.562	480.000		
Shape dünn-voll				
f distance	2.127	1.000	2.742	0.098
f position	1.643	1.000	2.118	0.146
signal	224.084	3.000	96.299	0.000
stimulus	0.236	1.000	0.305	0.581
f distance:f position	1.221	1.000	1.574	0.210
f distance:signal	38.631	3.000	16.601	0.000
f position:signal	3.428	3.000	1.473	0.221
f distance:stimulus	0.096	1.000	0.123	0.726
f position:stimulus	0.002	1.000	0.003	0.960
signal:stimulus	1.428	3.000	0.614	0.606
f distance:f position:signal	0.131	3.000	0.056	0.982
f distance:f position:stimulus	0.236	1.000	0.305	0.581
f distance:signal:stimulus	1.787	3.000	0.768	0.512
f position:signal:stimulus	0.881	3.000	0.379	0.769
f distance:f position:signal:stimulus	1.678	3.000	0.721	0.540
Residuals	372.313	480.000		
Shape dunkel-hell				
f distance	13.133	1.000	21.850	0.000
f position	0.008	1.000	0.013	0.909
signal	265.164	3.000	147.058	0.000
stimulus	0.383	1.000	0.637	0.425
f distance:f position	0.945	1.000	1.573	0.210
f distance:signal	18.695	3.000	10.368	0.000
f position:signal	1.352	3.000	0.750	0.523
f distance:stimulus	0.195	1.000	0.325	0.569
f position:stimulus	1.758	1.000	2.925	0.088
signal:stimulus	0.164	3.000	0.091	0.965
f distance:f position:signal	0.820	3.000	0.455	0.714
f distance:f position:stimulus	0.195	1.000	0.325	0.569
f distance:signal:stimulus	0.695	3.000	0.386	0.763
f position:signal:stimulus	0.914	3.000	0.507	0.678
f distance:f position:signal:stimulus	1.195	3.000	0.663	0.575
Residuals	288.500	480.000		
Shape kein-viel Bass				
f distance	21.125	1.000	38.067	0.000
f position	0.781	1.000	1.408	0.236
signal	357.008	3.000	214.439	0.000

continued on next page

continued from previous page

Shape	Sum Sq	Df	F value	Pr(> F)
stimulus	2.820	1.000	5.082	0.025
f distance:f position	1.758	1.000	3.168	0.076
f distance:signal	26.453	3.000	15.889	0.000
f position:signal	2.766	3.000	1.661	0.175
f distance:stimulus	0.281	1.000	0.507	0.477
f position:stimulus	3.125	1.000	5.631	0.018
signal:stimulus	3.977	3.000	2.389	0.068
f distance:f position:signal	1.227	3.000	0.737	0.530
f distance:f position:stimulus	0.633	1.000	1.140	0.286
f distance:signal:stimulus	3.766	3.000	2.262	0.080
f position:signal:stimulus	1.516	3.000	0.910	0.436
f distance:f position:signal:stimulus	1.570	3.000	0.943	0.420
Residuals	266.375	480.000		
Shape kein-viel Höhen				
f distance	3.781	1.000	7.435	0.007
f position	0.633	1.000	1.244	0.265
signal	127.656	3.000	83.666	0.000
stimulus	2.531	1.000	4.977	0.026
f distance:f position	0.070	1.000	0.138	0.710
f distance:signal	17.125	3.000	11.224	0.000
f position:signal	2.336	3.000	1.531	0.206
f distance:stimulus	0.281	1.000	0.553	0.457
f position:stimulus	0.008	1.000	0.015	0.901
signal:stimulus	1.312	3.000	0.860	0.462
f distance:f position:signal	0.461	3.000	0.302	0.824
f distance:f position:stimulus	0.945	1.000	1.859	0.173
f distance:signal:stimulus	0.312	3.000	0.205	0.893
f position:signal:stimulus	1.211	3.000	0.794	0.498
f distance:f position:signal:stimulus	0.711	3.000	0.466	0.706
Residuals	244.125	480.000		
Shape leise-laut				
f distance	3.781	1.000	8.843	0.003
f position	1.125	1.000	2.631	0.105
signal	15.336	3.000	11.955	0.000
stimulus	0.281	1.000	0.658	0.418
f distance:f position	0.383	1.000	0.895	0.345
f distance:signal	33.281	3.000	25.944	0.000
f position:signal	2.688	3.000	2.095	0.100
f distance:stimulus	0.945	1.000	2.211	0.138
f position:stimulus	0.195	1.000	0.457	0.499

continued on next page

continued from previous page

Shape	Sum Sq	Df	F value	Pr($> F$)
signal:stimulus	2.781	3.000	2.168	0.091
f distance:f position:signal	0.711	3.000	0.554	0.646
f distance:f position:stimulus	1.125	1.000	2.631	0.105
f distance:signal:stimulus	3.648	3.000	2.844	0.037
f position:signal:stimulus	0.773	3.000	0.603	0.613
f distance:f position:signal:stimulus	0.438	3.000	0.341	0.796
Residuals	205.250	480.000		

Component	1	2	3	4	5	6	7	8	9	10	11	12
eng - breit	−0.282	−0.379	−0.298	−0.273	0.114	−0.034	−0.258	−0.106	0.302	0.654	−0.036	−0.021
links - rechts	−0.080	−0.318	0.079	0.810	0.368	−0.171	−0.222	−0.085	−0.048	0.000	−0.077	0.012
nah - weit	−0.135	−0.494	0.370	0.011	−0.093	0.085	0.409	0.520	0.363	−0.065	0.066	0.085
oben - unten	0.017	−0.336	−0.295	0.251	−0.814	0.155	−0.125	−0.084	−0.138	−0.068	0.001	−0.065
trocken - räumlich	−0.253	−0.292	−0.219	−0.193	0.047	−0.594	0.438	−0.308	−0.199	−0.287	−0.044	−0.036
klare Ortung - diffus	−0.308	−0.273	0.017	−0.316	0.192	0.204	−0.541	0.177	−0.241	−0.506	0.114	−0.028
leise - laut	−0.083	0.258	−0.689	0.162	0.073	−0.153	−0.006	0.507	0.301	−0.211	−0.008	0.061
dumpf - klar	0.363	−0.204	−0.229	0.012	0.229	0.296	0.142	−0.348	0.333	−0.217	0.538	−0.214
dünn - voll	−0.370	0.051	−0.193	0.103	0.125	0.529	0.320	−0.195	−0.175	0.024	−0.019	0.587
dunkel - hell	0.404	−0.242	−0.140	−0.128	0.160	0.226	0.047	−0.048	0.088	−0.173	−0.789	−0.020
kein - viel Bass	−0.427	0.150	−0.027	0.117	0.064	0.315	0.243	0.021	−0.074	0.045	−0.160	−0.765
kein - viel Höhen	0.341	−0.219	−0.217	−0.036	0.195	0.050	0.182	0.398	−0.643	0.317	0.187	−0.083
standard deviation	3.781	1.593	1.249	0.942	0.912	0.789	0.699	0.624	0.516	0.408	0.269	0.217

Correlation	breit	rechts	weit	oben	räuml.	diffus	laut	klar	voll	hell	Bass	Höhen
eng-breit	1.000	0.121	0.219	0.129	0.432	0.495	0.118	−0.178	0.340	−0.220	0.303	−0.200
links-rechts	0.121	1.000	0.191	0.060	0.106	0.107	−0.037	−0.006	0.084	−0.079	0.084	−0.021
nah-weit	0.219	0.191	1.000	0.113	0.206	0.246	−0.268	−0.129	0.089	−0.075	0.148	−0.062
unten-oben	0.129	0.060	0.113	1.000	0.068	−0.003	0.034	0.078	0.001	0.083	−0.088	0.074
trocken-räumlich	0.432	0.106	0.206	0.068	1.000	0.270	0.088	−0.224	0.267	−0.267	0.260	−0.167
klare Ortung-diffus	0.495	0.107	0.246	−0.003	0.270	1.000	−0.026	−0.324	0.351	−0.287	0.368	−0.258
leise-laut	0.118	−0.037	−0.268	0.034	0.088	−0.026	1.000	−0.063	0.180	−0.127	0.184	−0.011
dumpf-klar	−0.178	−0.006	−0.129	0.078	−0.224	−0.324	−0.063	1.000	−0.306	0.689	−0.523	0.477
dünn-voll	0.340	0.084	0.089	0.001	0.267	0.351	0.180	−0.306	1.000	−0.442	0.729	−0.361
dunkel-hell	−0.220	−0.079	−0.075	0.083	−0.267	−0.287	−0.127	0.689	−0.442	1.000	−0.615	0.588
kein-viel Bass	0.303	0.084	0.148	−0.088	0.260	0.368	0.184	−0.523	0.729	−0.615	1.000	−0.503
kein-viel Höhen	−0.200	−0.021	−0.062	0.074	−0.167	−0.258	−0.011	0.477	−0.361	0.588	−0.503	1.000

Table E.4: Results of the principal components analysis and the correlation table, for the shape set

Table E.5: Detailed ANOVA results for the Size set

Size	Sum Sq	Df	F value	Pr(> F)
Size eng-breit				
f distance	0.264	1.000	0.239	0.625
f position	0.977	1.000	0.883	0.348
signal	111.962	4.000	25.295	0.000
stimulus	8.789	1.000	7.943	0.005
f distance:f position	0.077	1.000	0.069	0.793
f distance:signal	9.588	4.000	2.166	0.071
f position:signal	3.875	4.000	0.875	0.478
f distance:stimulus	1.502	1.000	1.357	0.245
f position:stimulus	0.264	1.000	0.239	0.625
signal:stimulus	14.312	4.000	3.234	0.012
f distance:f position:signal	1.712	4.000	0.387	0.818
f distance:f position:stimulus	1.702	1.000	1.538	0.215
f distance:signal:stimulus	0.475	4.000	0.107	0.980
f position:signal:stimulus	3.087	4.000	0.698	0.594
f distance:f position:signal:stimulus	2.962	4.000	0.669	0.613
Residuals	663.937	600.000		
Size links-rechts				
f distance	3.906	1.000	6.162	0.013
f position	4.900	1.000	7.729	0.006
signal	23.834	4.000	9.399	0.000
stimulus	0.100	1.000	0.158	0.691
f distance:f position	16.900	1.000	26.658	0.000
f distance:signal	12.703	4.000	5.009	0.001
f position:signal	66.553	4.000	26.245	0.000
f distance:stimulus	0.225	1.000	0.355	0.552
f position:stimulus	0.306	1.000	0.483	0.487
signal:stimulus	0.728	4.000	0.287	0.886
f distance:f position:signal	142.647	4.000	56.252	0.000
f distance:f position:stimulus	0.006	1.000	0.010	0.921
f distance:signal:stimulus	0.384	4.000	0.152	0.962
f position:signal:stimulus	6.459	4.000	2.547	0.038
f distance:f position:signal:stimulus	1.416	4.000	0.558	0.693
Residuals	380.375	600.000		
Size nah-weit				
f distance	0.352	1.000	0.512	0.474
f position	0.689	1.000	1.004	0.317

continued on next page

continued from previous page

Size	Sum Sq	Df	F value	Pr(> *F*)
signal	6.719	4.000	2.448	0.045
stimulus	0.127	1.000	0.184	0.668
f distance:f position	0.452	1.000	0.658	0.418
f distance:signal	34.000	4.000	12.388	0.000
f position:signal	1.069	4.000	0.389	0.816
f distance:stimulus	0.564	1.000	0.822	0.365
f position:stimulus	0.189	1.000	0.276	0.600
signal:stimulus	10.381	4.000	3.782	0.005
f distance:f position:signal	1.650	4.000	0.601	0.662
f distance:f position:stimulus	0.039	1.000	0.057	0.811
f distance:signal:stimulus	2.975	4.000	1.084	0.364
f position:signal:stimulus	0.319	4.000	0.116	0.977
f distance:f position:signal:stimulus	1.938	4.000	0.706	0.588
Residuals	411.687	600.000		
Size unten-oben				
f distance	2.889	1.000	5.830	0.016
f position	2.139	1.000	4.317	0.038
signal	5.734	4.000	2.893	0.022
stimulus	2.889	1.000	5.830	0.016
f distance:f position	0.264	1.000	0.533	0.466
f distance:signal	4.134	4.000	2.086	0.081
f position:signal	0.759	4.000	0.383	0.821
f distance:stimulus	0.014	1.000	0.028	0.866
f position:stimulus	0.352	1.000	0.709	0.400
signal:stimulus	1.134	4.000	0.572	0.683
f distance:f position:signal	1.572	4.000	0.793	0.530
f distance:f position:stimulus	0.564	1.000	1.138	0.286
f distance:signal:stimulus	1.759	4.000	0.888	0.471
f position:signal:stimulus	0.734	4.000	0.371	0.830
f distance:f position:signal:stimulus	0.022	4.000	0.011	1.000
Residuals	297.312	600.000		
Size trocken-räumlich				
f distance	0.264	1.000	0.194	0.660
f position	7.014	1.000	5.140	0.024
signal	59.713	4.000	10.941	0.000
stimulus	25.202	1.000	18.470	0.000
f distance:f position	1.502	1.000	1.100	0.295
f distance:signal	9.431	4.000	1.728	0.142
f position:signal	1.869	4.000	0.342	0.849
f distance:stimulus	0.039	1.000	0.029	0.866

continued on next page

continued from previous page

Size	Sum Sq	Df	F value	Pr(> F)
f position:stimulus	0.977	1.000	0.716	0.398
signal:stimulus	2.806	4.000	0.514	0.725
f distance:f position:signal	2.288	4.000	0.419	0.795
f distance:f position:stimulus	0.039	1.000	0.029	0.866
f distance:signal:stimulus	5.938	4.000	1.088	0.362
f position:signal:stimulus	1.875	4.000	0.344	0.849
f distance:f position:signal:stimulus	2.656	4.000	0.487	0.746
Residuals	818.688	600.000		
Size klare Ortung-diffus				
f distance	4.556	1.000	4.159	0.042
f position	1.406	1.000	1.284	0.258
signal	165.416	4.000	37.752	0.000
stimulus	7.656	1.000	6.989	0.008
f distance:f position	0.225	1.000	0.205	0.651
f distance:signal	39.897	4.000	9.105	0.000
f position:signal	1.859	4.000	0.424	0.791
f distance:stimulus	0.900	1.000	0.822	0.365
f position:stimulus	7.225	1.000	6.596	0.010
signal:stimulus	13.422	4.000	3.063	0.016
f distance:f position:signal	5.541	4.000	1.265	0.283
f distance:f position:stimulus	0.506	1.000	0.462	0.497
f distance:signal:stimulus	1.241	4.000	0.283	0.889
f position:signal:stimulus	4.666	4.000	1.065	0.373
f distance:f position:signal:stimulus	2.134	4.000	0.487	0.745
Residuals	657.250	600.000		
Size dumpf-klar				
f distance	30.189	1.000	44.389	0.000
f position	1.702	1.000	2.502	0.114
signal	454.741	4.000	167.158	0.000
stimulus	7.439	1.000	10.938	0.001
f distance:f position	1.139	1.000	1.675	0.196
f distance:signal	58.553	4.000	21.524	0.000
f position:signal	5.728	4.000	2.106	0.079
f distance:stimulus	0.264	1.000	0.388	0.533
f position:stimulus	0.002	1.000	0.002	0.962
signal:stimulus	11.022	4.000	4.052	0.003
f distance:f position:signal	2.478	4.000	0.911	0.457
f distance:f position:stimulus	0.564	1.000	0.829	0.363
f distance:signal:stimulus	1.322	4.000	0.486	0.746
f position:signal:stimulus	1.334	4.000	0.491	0.743

continued on next page

continued from previous page				
Size	Sum Sq	Df	F value	Pr($> F$)
f distance:f position:signal:stimulus	1.147	4.000	0.422	0.793
Residuals	408.062	600.000		
Size dünn-voll				
f distance	8.100	1.000	10.199	0.001
f position	5.256	1.000	6.619	0.010
signal	337.884	4.000	106.364	0.000
stimulus	4.900	1.000	6.170	0.013
f distance:f position	0.025	1.000	0.031	0.859
f distance:signal	19.447	4.000	6.122	0.000
f position:signal	2.572	4.000	0.810	0.519
f distance:stimulus	0.156	1.000	0.197	0.658
f position:stimulus	0.000	1.000	0.000	1.000
signal:stimulus	4.022	4.000	1.266	0.282
f distance:f position:signal	1.459	4.000	0.459	0.766
f distance:f position:stimulus	1.406	1.000	1.771	0.184
f distance:signal:stimulus	2.859	4.000	0.900	0.463
f position:signal:stimulus	1.047	4.000	0.330	0.858
f distance:f position:signal:stimulus	3.359	4.000	1.058	0.377
Residuals	476.500	600.000		
Size dunkel-hell				
f distance	26.814	1.000	41.512	0.000
f position	1.502	1.000	2.325	0.128
signal	362.119	4.000	140.152	0.000
stimulus	1.314	1.000	2.034	0.154
f distance:f position	0.077	1.000	0.119	0.731
f distance:signal	49.913	4.000	19.318	0.000
f position:signal	0.725	4.000	0.281	0.891
f distance:stimulus	0.014	1.000	0.022	0.883
f position:stimulus	0.689	1.000	1.067	0.302
signal:stimulus	1.256	4.000	0.486	0.746
f distance:f position:signal	6.369	4.000	2.465	0.044
f distance:f position:stimulus	0.127	1.000	0.196	0.658
f distance:signal:stimulus	0.775	4.000	0.300	0.878
f position:signal:stimulus	2.100	4.000	0.813	0.517
f distance:f position:signal:stimulus	1.319	4.000	0.510	0.728
Residuals	387.562	600.000		
Size kein-viel Bass				
f distance	24.025	1.000	37.785	0.000
f position	0.625	1.000	0.983	0.322
signal	516.681	4.000	203.151	0.000
				continued on next page

continued from previous page

Size	Sum Sq	Df	F value	Pr(> F)
stimulus	6.006	1.000	9.446	0.002
f distance:f position	0.306	1.000	0.482	0.488
f distance:signal	43.694	4.000	17.180	0.000
f position:signal	2.375	4.000	0.934	0.444
f distance:stimulus	3.025	1.000	4.758	0.030
f position:stimulus	1.225	1.000	1.927	0.166
signal:stimulus	11.463	4.000	4.507	0.001
f distance:f position:signal	0.788	4.000	0.310	0.872
f distance:f position:stimulus	0.156	1.000	0.246	0.620
f distance:signal:stimulus	1.975	4.000	0.777	0.541
f position:signal:stimulus	1.994	4.000	0.784	0.536
f distance:f position:signal:stimulus	0.906	4.000	0.356	0.840
Residuals	381.500	600.000		
Size kein-viel Höhen				
f distance	5.256	1.000	9.108	0.003
f position	0.756	1.000	1.310	0.253
signal	185.366	4.000	80.303	0.000
stimulus	0.006	1.000	0.011	0.917
f distance:f position	0.006	1.000	0.011	0.917
f distance:signal	31.166	4.000	13.501	0.000
f position:signal	0.228	4.000	0.099	0.983
f distance:stimulus	0.506	1.000	0.877	0.349
f position:stimulus	1.406	1.000	2.437	0.119
signal:stimulus	3.322	4.000	1.439	0.220
f distance:f position:signal	1.916	4.000	0.830	0.506
f distance:f position:stimulus	0.756	1.000	1.310	0.253
f distance:signal:stimulus	0.259	4.000	0.112	0.978
f position:signal:stimulus	0.859	4.000	0.372	0.828
f distance:f position:signal:stimulus	2.384	4.000	1.033	0.389
Residuals	346.250	600.000		
Size leise-laut				
f distance	0.077	1.000	0.171	0.680
f position	1.314	1.000	2.932	0.087
signal	46.413	4.000	25.887	0.000
stimulus	1.502	1.000	3.350	0.068
f distance:f position	0.564	1.000	1.258	0.262
f distance:signal	25.119	4.000	14.010	0.000
f position:signal	1.100	4.000	0.614	0.653
f distance:stimulus	0.564	1.000	1.258	0.262
f position:stimulus	0.189	1.000	0.422	0.516

continued on next page

continued from previous page

Size	Sum Sq	Df	F value	Pr($> F$)
signal:stimulus	2.819	4.000	1.572	0.180
f distance:f position:signal	0.631	4.000	0.352	0.843
f distance:f position:stimulus	0.352	1.000	0.784	0.376
f distance:signal:stimulus	1.100	4.000	0.614	0.653
f position:signal:stimulus	1.319	4.000	0.736	0.568
f distance:f position:signal:stimulus	0.625	4.000	0.349	0.845
Residuals	268.938	600.000		

Components	1	2	3	4	5	6	7	8	9	10	11	12
eng - breit	−0.266	0.394	0.235	−0.185	0.261	−0.013	0.412	−0.223	0.514	−0.353	−0.091	−0.026
links - rechts	−0.109	0.282	−0.277	0.766	0.413	−0.143	0.125	−0.023	−0.194	−0.002	0.002	0.020
nah - weit	−0.014	0.419	−0.627	−0.051	−0.228	0.391	−0.268	0.187	0.335	−0.029	−0.031	−0.051
oben - unten	0.021	0.500	0.168	0.110	−0.685	−0.412	0.191	0.026	−0.123	0.124	0.026	0.032
trocken - räumlich	−0.211	0.387	0.089	−0.363	0.380	−0.379	−0.552	0.094	−0.159	0.187	0.057	−0.031
klare Ortung - diffus	−0.330	0.174	−0.056	−0.252	0.056	0.458	0.398	−0.128	−0.428	0.446	0.156	−0.007
leise - laut	−0.227	0.076	0.575	0.305	−0.028	0.383	−0.158	0.568	0.098	0.045	0.024	−0.105
dumpf - klar	0.403	0.099	0.135	0.116	0.083	0.051	−0.053	−0.291	0.268	0.327	0.468	−0.550
dünn - voll	−0.364	−0.079	0.121	0.222	−0.187	0.119	−0.372	−0.531	0.179	0.062	0.283	0.461
dunkel - hell	0.386	0.175	0.188	0.057	0.115	0.167	−0.068	−0.191	0.116	0.436	−0.646	0.280
kein - viel Bass	−0.405	−0.133	−0.015	0.100	−0.188	−0.009	−0.178	−0.313	−0.091	−0.015	−0.495	−0.624
kein - viel Höhen	0.323	0.298	0.191	−0.021	−0.014	0.335	−0.202	−0.255	−0.477	−0.568	0.024	−0.034
standard deviation	4.348	1.582	1.168	0.931	0.864	0.684	0.605	0.529	0.420	0.408	0.253	0.208

Correlation	breit	rechts	weit	oben	räuml.	diffus	laut	klar	voll	hell	Bass	Höhen
eng-breit	1.000	0.180	0.047	0.163	0.449	0.481	0.313	−0.344	0.313	−0.290	0.311	−0.179
links-rechts	0.180	1.000	0.190	0.057	0.128	0.113	0.098	−0.107	0.143	−0.111	0.137	−0.107
nah-weit	0.047	0.190	1.000	0.179	0.119	0.155	−0.167	−0.050	−0.036	−0.021	−0.027	0.079
unten-oben	0.163	0.057	0.179	1.000	0.104	−0.002	0.078	0.082	−0.024	0.105	−0.050	0.184
trocken-räumlich	0.449	0.128	0.119	0.104	1.000	0.314	0.181	−0.299	0.221	−0.224	0.240	−0.133
klare Ortung-diffus	0.481	0.113	0.155	−0.002	0.314	1.000	0.271	−0.529	0.405	−0.446	0.484	−0.333
leise-laut	0.313	0.098	−0.167	0.078	0.181	0.271	1.000	−0.300	0.405	−0.225	0.333	−0.159
dumpf-klar	−0.344	−0.107	−0.050	0.082	−0.299	−0.529	−0.300	1.000	−0.512	0.748	−0.681	0.573
dünn-voll	0.313	0.143	−0.036	−0.024	0.221	0.405	0.405	−0.512	1.000	−0.530	0.731	−0.431
dunkel-hell	−0.290	−0.111	−0.021	0.105	−0.224	−0.446	−0.225	0.748	−0.530	1.000	−0.659	0.607
kein-viel Bass	0.311	0.137	−0.027	−0.050	0.240	0.484	0.333	−0.681	0.731	−0.659	1.000	−0.550
kein-viel Höhen	−0.179	−0.107	0.079	0.184	−0.133	−0.333	−0.159	0.573	−0.431	0.607	−0.550	1.000

Table E.6: Results of the principal components analysis and the correlation table, for the size set

Curriculum Vitae

Name:	Marije Alberdina Johanna Baalman
Birth:	7th of March, 1978 in Pingjum, Friesland, The Netherlands
Nationality:	Dutch

Education

1982-1984	Kleuterschool *"Op'e Drompel"*, Pingjum
1984-1990	Openbare Basisschool *"It Leech"*, Pingjum
1990-1996	Gymnasium, Regionale Scholengemeenschap *"Magister Alvinus"* te Sneek
1996-2002	Applied Physics, Technische Universiteit Delft Specialisation: Acoustics Subject: Perception of Spaciousness in Concert Halls
2001-2002	Sonology course, Royal Conservatory of The Hague
2002-2007	Doctoral student at the Technische Universität Berlin, Germany
2007-2008	Post-Doc Fellowship at Concordia University, Montréal, Canada

Grants

2002-2003	VSB Beurs, Fortis Bank, the Netherlands
2003-2004	Deutsche Akademische Austauschdienst (DAAD)
2005-2007	Nachwuchs Förderungsgesetz (NaFöG), Berlin

Research

2002-2007	Wave Field Synthesis, KW, TU Berlin Development of the *sWONDER* software
2002-2008	Sensor interfaces for live perfomance, personal research
2004-2005	Active noise control, ITA, TU Berlin
2005-2008	*Predictive Sonobotanics*, IPSO-FACTO in cooperation with Alberto de Campo
2005-2008	"Schwelle" - Use of sensor interfaces for Dance in cooperation with Chris Salter, Concordia University, Montréal
2007-2008	Use of networked wireless technologies in performance Concordia University and McGill University, Montréal

Since 2002 active as sound artist and performer, and providing technical support for artists.