



ADAPTION DER LIGHTMAP ALS DATENSTRUKTUR IN DER RAUMAKUSTISCHEN SIMULATION

Matthias Fottner, David Keller

BACHELORARBEIT

angefertigt im Rahmen des Seminars

Elektro- und Raumakustik

Betreuer

Dipl.-Ing. Dr.techn. Werner Weselak

Signal Processing and Speech Communication Laboratory

Institutsleiter: Univ.-Prof. Dipl.-Ing. Dr.techn. Gernot Kubin

Technische Universität Graz

Graz, 21. Dezember 2022

Literatur

- [1] R. Schärfig, „Efficient multi-bounce lightmap creation using GPU forward mapping,“ Diss., Università della Svizzera italiana Lugano, 2016.
- [2] „Lehrveranstaltung Raumakustik,“ Technische Universität Graz.
- [3] M. Vorländer, *Auralization: fundamentals of acoustics, modelling, simulation, algorithms and acoustic virtual reality; 2nd ed.* Ser. RWTHedition. Dordrecht: Springer Berlin, Heidelberg, 2020. DOI: 10.1007/978-3-540-48830-9.
- [4] B. Keinert, M. Innmann, M. Sängler und M. Stamminger, „Spherical Fibonacci Mapping,“ *ACM Transactions on Graphics*, Jg. 34, S. 1–7, Okt. 2015. DOI: 10.1145/2816795.2818131.
- [5] Scratchapixel. „Barycentric Coordinates.“ (12. Sep. 2022), Adresse: <https://www.scratchapixel.com/lessons/3d-basic-rendering/ray-tracing-rendering-a-triangle/barycentric-coordinates> (besucht am 12.09.2022).
- [6] T. Möller und B. Trumbore, „Fast, Minimum Storage Ray-Triangle Intersection,“ *Journal of Graphics Tools*, Jg. 2, Nr. 1, S. 21–28, 1997. DOI: 10.1080/10867651.1997.10487468. eprint: <https://doi.org/10.1080/10867651.1997.10487468>. Adresse: <https://doi.org/10.1080/10867651.1997.10487468>.
- [7] K. W. Beick, „Analyzing Performance of Bounding Volume Hierarchies For Ray Tracing,“ *Department of Computer and Information Science at Robert D. Clark Honors College*, 2014.
- [8] Z. Xiangyang, C. Ke’an und S. Jincai, „On the accuracy of the ray-tracing algorithms based on various sound receiver models,“ *Applied Acoustics*, Jg. 64, Nr. 4, S. 433–441, 2003, ISSN: 0003-682X. DOI: [https://doi.org/10.1016/S0003-682X\(02\)00108-1](https://doi.org/10.1016/S0003-682X(02)00108-1). Adresse: <https://www.sciencedirect.com/science/article/pii/S0003682X02001081>.
- [9] J. M. Cardoso, J. G. F. Coutinho und P. C. Diniz, „Chapter 2 - High-performance embedded computing,“ in *Embedded Computing for High Performance*, J. M. Cardoso, J. G. F. Coutinho und P. C. Diniz, Hrsg., Boston: Morgan Kaufmann, 2017, S. 17–56, ISBN: 978-0-12-804189-5. DOI: <https://doi.org/10.1016/B978-0-12-804189-5.00002-8>. Adresse: <https://www.sciencedirect.com/science/article/pii/B9780128041895000028>.
- [10] V. Falco. „A Collection of Useful C++ Classes for Digital Signal Processing.“ (9. März 2017), Adresse: <https://github.com/vinniefalco/DSPFilters> (besucht am 14.11.2022).
- [11] B. Foundation. „Reusing Existing Materials.“ (6. Aug. 2022), Adresse: <https://docs.blender.org/manual/en/latest/render/materials/assignment.html#reusing-existing-materials>.
- [12] R. Hofer, „Analyse des modalen Schallfeldes zur Untersuchung der Funktionsweise von Kantenabsorbieren,“ Masterarb. Technische Universität Graz, 2022.
- [13] Institute of Technical Acoustics of the RWTH Aachen University. „ITA-Toolbox.“ (27. Okt. 2022), Adresse: <https://www.ita-toolbox.org/>.
- [14] „Akustik - Messung der Schallabsorption in Hallräumen,“ EN ISO 354, Dez. 2003.

Inhaltsverzeichnis

Literatur	III
1 Einleitung	1
1.1 Motivation	1
1.2 Geometrische Raumakustik	1
1.3 Stochastisches Raytracing	2
1.4 Raytracing vs. FEM-Simulation	2
2 Softwareentwicklung	3
2.1 Erstellen einer Kugelschallquelle	3
2.2 Erstellen des Raummodells	4
2.2.1 .obj-Datei Einführung	4
2.2.2 Triangulate Faces	4
2.2.3 Flatshading	4
2.2.4 Material	6
2.3 Raytracing Programmierschnittstelle	6
2.4 Soundmap	8
2.4.1 Übersicht	8
2.4.2 Berechnung der Parameter eines Soundmap-Eintrags	9
2.4.3 Auswertung an der Empfängerposition	10
2.4.4 Erstellen der oktavbandabhängigen Impulsantworten	12
2.4.5 Erstellen der breitbandigen Impulsantwort	14
3 Leitfaden zur ARTWA Software	15
3.1 Vorkonfiguration	15
3.1.1 Addon für Blender	15
3.1.2 Anwendung mit einem beliebigen CAD-Programm	16
3.1.3 Zuweisung von Materialien	16
3.2 ARTWA	17
3.2.1 Import	18
3.2.2 Simulation	18
3.2.3 Render Settings	19
3.2.4 Audio Settings	20
3.2.5 Impulse Response	22
3.2.6 Materiallegende	23
4 Vergleich mit kommerzieller Software	25
4.1 Hallraum des Labors für Bauphysik an der TU Graz	25
4.1.1 Raumeigenschaften	25
4.1.2 Simulationseigenschaften	26
4.1.3 Abklingkurven	27
4.2 Absorbierender Raum	28
4.2.1 Raumeigenschaften	28
4.2.2 Simulationseigenschaften	28
4.2.3 Abklingkurven	29
4.3 Konzertsaal mit gemischten Materialien	30
4.3.1 Raumeigenschaften	30
4.3.2 Simulationseigenschaften	31
4.3.3 Abklingkurven	31

5	Diskussion	33
5.1	Early Decay Time	33
5.2	Auswirkung der Soundmap	33
5.3	Systematische Fehler	34
6	Zusammenfassung und Ausblick	35
6.1	Zusammenfassung	35
6.2	Ausblick	36
6.2.1	Abbruchkriterium	36
6.2.2	Beugung und Scattering	36
6.2.3	Hybride Modelle	36
6.2.4	Projektdatei	36

1

Einleitung

1.1 Motivation

Aktuell am Markt angebotene Raumakustik Software bietet Anlass zur Verbesserung. So kann bei manchen Herstellern lediglich vor der Simulation eine begrenzte Anzahl (z.B.: 100) an Empfängerpositionen angegeben werden, für welche anschließend die Impulsantwort berechnet wird. Da die während einer Simulation angestellten Kalkulationen sehr zeitintensiv sind, kann ein Raum niemals vollständig behandelt und somit an jeder Stelle “abgehört” werden. Die Benutzeroberflächen sind in vielen Fällen veraltet und zeigen sich daher als unintuitiv. 3D-Visualisierungen von den angestellten Berechnungen werden nur in seltenen Fällen angeboten, wobei diesen einiges an Erkenntnis abgewonnen werden könnte.

In der hier vorgestellten Software werden oben genannte Punkte behandelt. Im Bereich der Raumakustik gibt es verschiedene Modelle, um die Schallausbreitung näher zu beschreiben (Wellentheoretische Raumakustik, Statistische Raumakustik, Geometrische Raumakustik). ARTWA (“Almost Realtime Walkthrough Auralisation”) arbeitet dabei mit der auf Schallstrahlen basierten Theorie, welche der geometrischen Raumakustik angehört. Für die Strahlensimulation selbst, können bereits vorhandene, äußerst effiziente, Ray-Tracing-Algorithmen adaptiert und verwendet werden. Um die oben beschriebene Empfänger-Ortsunabhängigkeit zu erzielen, wird auf eine eigens kreierte “Soundmap” gesetzt, welche das Pendant zur in der Computergrafik eingesetzten Datenstruktur für Simulationen von fixen Lichtquellen (Lightmap) darstellt [1]. Durch diese vollständige Entkopplung von der Empfängerposition, muss der wirklich rechenzeitintensive Part einer Simulation (3 - 10 Minuten) ausschließlich einmal durchgeführt werden. Ein in 3D dargestellter Raum kann somit durchschritten werden, wobei sich mit jedem Positionswechsel die akustische Wahrnehmung innerhalb von einigen Sekunden (bis maximal 2 Minuten, siehe Kapitel 5.2) verändert. Diese Auralisation lässt sich über die Faltung der ortsabhängigen Impulsantwort mit der jeweiligen Beispiel-Audio-Datei durchführen.

1.2 Geometrische Raumakustik

In einem Medium mit einer an jedem Ort gleichen Fortpflanzungsgeschwindigkeit, erfolgt die Wellenausbreitung immer auf dem geometrisch kürzesten Weg. Diese Tatsache wurde bereits in der Optik festgestellt und so kann Lichtausbreitung in Form von Lichtstrahlengeraden dargestellt werden. Auf dieser Grundlage aufbauend, wird die geometrische Raumakustik formuliert. Von einer Kugelquelle in einem Raum gehen Schallstrahlen jeweils mit gleichverteilten Energieanteilen (omnidirektional) geradlinig aus, bis jene auf ein “Hindernis” treffen. Der Schallstrahl kann in diesem Fall entweder reflektiert oder abgeschattet werden [2]. Je nach Beschaffenheit der Oberfläche des Hindernisses kommt es zu verschiedenen Reflexionsverhalten. Um eine reguläre Reflexion erwarten zu können (Einfallswinkel = Reflexionswinkel, Einfallstrahl, -lot und reflektierter Strahl liegen in einer Ebene) ist eine, in Bezug auf die Wellenlänge, große reflektierende Fläche Voraussetzung. ARTWA beachtet die Struktur der Oberflächen allerdings nicht

und arbeitet ausschließlich mit regulären Reflexionen.

1.3 Stochastisches Raytracing

Beim stochastischen Raytracing wird Energie mithilfe einer großen Anzahl (z.B.: 10^6) von sogenannten Partikeln transportiert. Diese werden von der Schallquelle, an einer gewissen Position im Raum, zum gleichen Zeitpunkt gleichmäßig in alle Richtungen emittiert. Die Richtwirkung einer solchen kann über die richtungsspezifischen Anfangsenergien der Teilchen berücksichtigt werden. Sollte ein Partikel eine Geometrieoberfläche treffen, so wird diesem, je nach Absorptionsgrad, Energie entnommen und es wird von der Begrenzungsfläche reflektiert. Für die Simulation ergeben sich daraus 2 mögliche Abbruchkriterien:

- ein Partikel-Energieminimum von e_{\min} wird erreicht
- ein Ray überschreitet eine maximale Reisezeit von t_{\max}

Bei der Auswertung am Empfänger wird anschließend jene Anzahl der Partikel ermittelt bzw. deren Energie aufaddiert, welche innerhalb eines Zeitpunktes eingefallen sind. Jedes Partikel (Ray) besitzt ein Energiepaket, welches Energie nur aufgrund von Absorption an der Wandfläche verliert. Die bei Kugelwellenausbreitung übliche Entfernungsdämpfung wird somit bei der Modellierung des Rays nicht beachtet. Das Abstandsgesetz ($E \sim \frac{1}{r^2}$) wird allerdings implizit, wie in Abb. 1.1 ersichtlich, durch die radiale Quellenabstrahlung und der damit verbundenen entfernungsabhängigen Strahlendichte berücksichtigt. Diese Art von Strahlenerkennung kann mit einem Mikrophon mit Kugelcharakteristik verglichen werden und beinhaltet daher keine binauralen Informationen. [3]

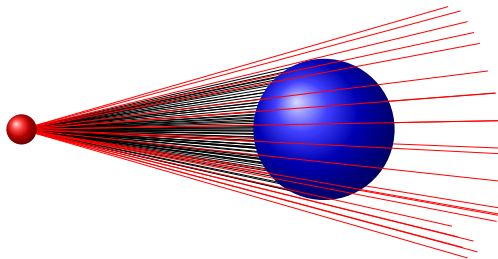


Abbildung 1.1: Am Empfänger detektierte Energie beim stochastischen Raytracing ausgehend von einer Kugelquelle [3].

1.4 Raytracing vs. FEM-Simulation

Zur FEM-Simulation muss das Raumvolumen in ausreichend kleine Volumina (sog. “Voxel”) unterteilt werden, um die physikalischen Eigenschaften der Schallausbreitung zu simulieren. Bei großen Räumen, wie z.B. Konzertsälen, entsteht somit sehr großer Speicherbedarf. Weiters muss für jeden Voxel das Verhalten berechnet werden. Ein Strahlenmodell hätte den Vorteil, nur dort Daten zu speichern, wo eine Reflexion stattgefunden hat. So minimiert sich sowohl der Rechenaufwand, als auch der Speicherbedarf. Als Nachteil jedoch kann die physikalische Ungenauigkeit, vor allem im Tieftonbereich, genannt werden.

2

Softwareentwicklung

2.1 Erstellen einer Kugelschallquelle

Zum Simulieren einer Kugelschallquelle ist es notwendig, Schallstrahlen gleichverteilt in alle Raumrichtungen auszusenden. Um dies zu erreichen muss das Problem gelöst werden, n Punkte auf einer Kugeloberfläche gleichmäßig zu verteilen. Eine schnelle Methode hierfür ist das ‘‘Spherical Fibonacci Mapping’’ (vgl. [4]). Hierbei wird zuerst mithilfe des goldenen Schnitts $\Phi = (\sqrt{5} + 1)/2$ ein Fibonacci-Gitter im Einheitsquadrat $[0, 1)^2$ (Glg. 2.1, Abb. 2.1(a)) erzeugt:

$$x_i = \left[\frac{i}{\Phi} \right] \quad \text{mit } [a] = a - \lfloor a \rfloor; \quad i \in \mathbb{N} \mid 0 \leq i < n,$$

$$\lfloor a \rfloor = \max\{m \in \mathbb{Z} \mid m \leq a\}$$

$$y_i = \frac{i}{n} \tag{2.1}$$

Dieses Gitter kann nun innerhalb des Einheitskreises (Glg. 2.2, Abb. 2.1(b)) und weiters auf die Oberfläche der Einheitskugel (Glg. 2.3, Abb. 2.1(c)) transformiert werden.

(‘‘Definitions­menge’’) \longrightarrow (‘‘Ziel­menge’’) : (‘‘Ab­bildungs­vorschrift’’)

$$(x_i, y_i) \longrightarrow (\theta_i, r_i) : (2\pi x_i, \sqrt{y_i}) \tag{2.2}$$

$$(x_i, y_i) \longrightarrow (\theta_i, \phi_i, r_i) : (2\pi x_i, \arccos(1 - 2y_i), 1)$$

$$(\theta_i, \phi_i, r_i) \longrightarrow (x_i, y_i, z_i) : (\cos \theta_i \sin \phi_i, \sin \theta_i \sin \phi_i, \cos \phi_i) \tag{2.3}$$

Die Punktkoordinaten (x_i, y_i, z_i) können nun als Richtungsvektoren für den ersten Simulationsschritt der Strahlen verwendet werden. Der Ursprung der Strahlen ist dem im GUI eingestellten Wert für ‘‘source position’’ gleichzusetzen.

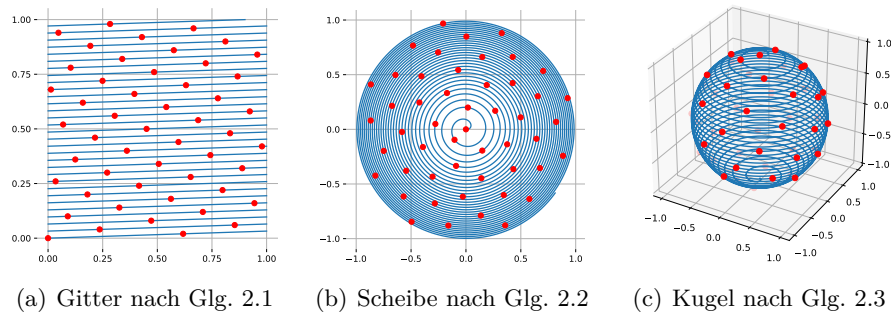


Abbildung 2.1: Verteilen von $n = 50$ Punkten auf der Einheitskugel.

2.2 Erstellen des Raummodells

Zum Erstellen eines Raummodells eignet sich jedes herkömmliche 3D-Modelling-Programm, da der Export als .obj-Datei zum Standard gehört. Hier wird die Verwendung von Blender empfohlen. Als umfangreiche Open Source Lösung bietet sie alle notwendigen Tools und wurde auch im Zuge dieser Arbeit verwendet.

2.2.1 .obj-Datei Einführung

Geometrieobjekte werden in der Computergrafik als sogenannte Netze (Meshes) gespeichert. Innerhalb einer solchen Datenstruktur werden Punkte (Vertices) und Normalen aufgelistet. Vertices werden durch deren Position definiert, Normalen beinhalten einen Richtungsvektor. Zusätzlich werden nun alle Flächen (Faces) gelistet, welche je Eckpunkt eine Referenz auf einen Vertex und eine Normale besitzen. Außerdem wird auf die zugehörige .mtl-Datei (Kapitel 2.2.4) verwiesen, in der Materialien des Meshes hinterlegt sind. Der Befehl `usemtl`, gefolgt von einem in der .mtl-Datei gelisteten Materialnamen, setzt für alle kommenden Faces das Material.

```
mtllib material_file.mtl      # Materialbibliothek (Dateiname.mtl)

v 1.000000 0.000000 0.000000  # Vertex 1 (v1) (x,y,z)
v 0.000000 1.000000 0.000000  # Vertex 2 (v2)
v 0.000000 0.000000 1.000000  # Vertex 3 (v3)

vn 0.000000 -1.000000 0.000000 # Normale 1 (n1) (x,y,z)
vn 0.000000 0.000000 -1.000000 # Normale 2 (n2)

usemtl Beton                  # Beispielmaterial
f 1//1 2//1 3//2             # Fläche 1 (gekrümmt) mit 3 Eckpunkten in folgendem Format:
                              # f v1//n1 v2//n1 v3//n2
```

Abbildung 2.2: Beispiel-.obj-Datei mit 3 Vertices, 2 Normalen und einer Dreiecksfläche. Diesem Dreieck wird das Material Beton zugewiesen.

Die in den Kapiteln 2.2.2 - 2.2.4 genannten Punkte sind beim Exportieren der .obj-Datei aus einem 3D-Modelling-Programm zu beachten.

2.2.2 Triangulate Faces

In 3D-Programmen kann es vorkommen, dass Flächen aus Polygonen bestehen. Der in ARTWA implementierte .obj-Datei-Parser erwartet allerdings ein reines Mesh aus Dreiecken. Daher müssen alle Polygone in Dreieckstrukturen zerlegt werden. Diese Option steht bei Blender der Exportfunktion mit dem Haken “Triangulate Faces” zur Verfügung bzw. wird von dem in Kapitel 3.1.1 vorgestellten Addon für Blender automatisch berücksichtigt. Alternativ kann das Mesh bereits rein aus Dreiecken modelliert werden (Blender-Funktion: Triangulate-Modifier).

2.2.3 Flatshading

Für eine präzise Simulation sollte beim Export “Flatshading” gewählt werden. Beim Flatshading wird innerhalb einer Fläche für jeden Eckpunkt auf die gleiche Normale referenziert. Dadurch ergeben sich sichtbare Kanten zwischen benachbarten Faces, deren Vertices nicht in einer Ebene liegen (Abbildung 2.4, links). Um ein Mesh aus nur wenigen Vertices trotzdem hochauflösend

erscheinen zu lassen, gibt es die Möglichkeit des Smoothshadings (Abbildung 2.4, rechts). Hierbei werden für die Eckpunkte eines Dreiecks auf unterschiedliche Normalen referenziert. Um die Flächennormale für einen beliebigen Punkt auf einem Dreieck zu erhalten, wird baryzentrisch interpoliert [5]. Dazu werden die Normalen der Eckpunkte mit u , v und w gewichtet (wobei $u + v + w = 1$), aufaddiert und normalisiert (Abbildung 2.3).

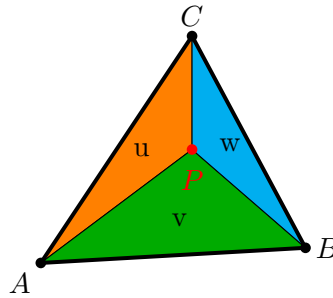


Abbildung 2.3: Baryzentrische Interpolation des Punktes P mithilfe der Flächeninhalte der Dreiecke APC , APB und BPC . Die Flächeninhalte u , v und w ergeben in Summe 1.

Obwohl diese baryzentrischen Normalen von ARTWA berechnet wurden, konnte bei solchen Meshes während der Simulation nicht immer auf einen Reflexionspunkt (Schnittpunkt mit der Geometrie) geschlossen werden. Dieses Problem ist auf numerische Ungenauigkeiten zurückzuführen. Daher wurde die Berechnung der baryzentrischen Normalen entfernt und Meshes mit Smoothshading können nicht behandelt werden.

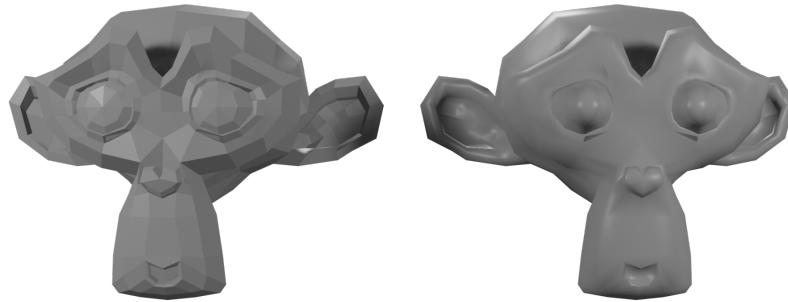


Abbildung 2.4: Flatshading (links) vs Smoothshading (rechts) bei gleicher Meshauflösung.

2.2.4 Material

Der Benutzer hat die Möglichkeit, Materialien in einer .csv-Datei zu definieren. Über das in Kapitel 3.1.1 beschriebene Addon werden die in der .csv-Datei gesetzten Parameter in eine Blendermaterialbibliothek übersetzt. Der modellierten Geometrie können anschließend die Materialien in Blender zugewiesen werden (Kapitel 3.1.3). Es folgt der Export über das Addon, welches die für ARTWA notwendige .obj-Datei mit einer namensgleichen .mtl-Datei erstellt. In der folgenden Tabelle wird der Zusammenhang zwischen der .csv-Datei, der .mtl-Datenstruktur, den Blenderbezeichnungen und den jeweiligen Parameterbedeutungen für ARTWA aufgeschlüsselt:

.csv- Bezeichnung	Blender- Bezeichnung	.mtl- Bezeichnung	Bedeutung für ARTWA
125	Metallic	Ka[0]	Absorptionsgrad für 125 Hz Oktavband
250	Specular	Ks	Absorptionsgrad für 250 Hz Oktavband
500	Alpha	d	Absorptionsgrad für 500 Hz Oktavband
1000	Emission (Red)	Ke[0]	Absorptionsgrad für 1000 Hz Oktavband
2000	Emission (Green)	Ke[1]	Absorptionsgrad für 2000 Hz Oktavband
4000	Emission (Blue)	Ke[2]	Absorptionsgrad für 4000 Hz Oktavband

.csv- Bezeichnung	Blender- Bezeichnung	.mtl- Bezeichnung	Bedeutung für ARTWA
R	Base Color (Red)	Kd[0]	Farbe Rot
G	Base Color (Green)	Kd[1]	Farbe Grün
B	Base Color (Blue)	Kd[2]	Farbe Blau

Tabelle 2.1: Zusammenhang der Materialparameter zwischen einer .mtl-Datei, Blender und der .csv-Datei aus Tabelle 3.1.

2.3 Raytracing Programmierschnittstelle

Das Embree API (Application Programming Interface) ist eine von Intel zur Verfügung gestellte Programmierschnittstelle, welche für Raytracing eingesetzt werden kann.

Dem `rtcscene`-Datentyp wird die Geometrie des Raums und das eingebrachte Mobiliar in Form eines triangulierten Meshs (Kapitel 2.2) hinzugefügt. Es werden nun Funktionen geboten, Strahlen mit einem gewissen Richtungsvektor innerhalb der Szene von einem davor definierten Punkt “auszuschicken”. Der `rtcrayhit`-Datentyp beinhaltet schlussendlich die ID des jeweiligen Dreiecks und implizit den dazugehörigen Punkt, bei dem der Strahl eintrifft. Der Richtungsvektor des Strahls wird durch Spiegelung an der Flächennormalen von diesem Punkt aus an dem getroffenen Dreieck reflektiert und der Vorgang beginnt erneut. Um den Schnittpunkt eines Strahls mit einer Geometrie ausfindig zu machen, kommen in der modernen Computergrafik verschiedene Algorithmen zum Einsatz. Embree greift hierbei auf den Möller-Trumbore-Algorithmus zurück [6].

Um eine Aussage über den Schnittpunkt treffen zu können, müsste prinzipiell jeder Strahl mit jedem einzelnen Dreieck geschnitten werden. Eine herkömmliche Geometrie eines Raums und dessen Mobiliar kann mehrere Millionen Dreiecke umfassen. Um eine Kugelquelle so akkurat wie möglich nachbilden zu können, sollte auch die Strahlenanzahl möglichst hoch angesetzt werden. Bei einer naiven Herangehensweise, bei der ein Strahl gegen jede Fläche getestet werden muss, entsteht im worst case ein linearer Aufwand.

Intel hingegen setzt dabei auf das sogenannte BVH (bounding volume hierarchy) Modell, welches folgendermaßen aufgebaut ist: Eine Geometrie wird rekursiv in immer kleinere quaderförmige Teilvolumina zerschnitten. Diese werden anschließend innerhalb einer Baumstruktur gespeichert, wobei die Bounding Box (Abb. 2.6) der gesamten Geometrie den Wurzelknoten des Baums repräsentiert. Anschließend werden weitere Bounding Boxes für Objekte innerhalb des inneren Knotens ermittelt und erneut Begrenzungsvolumina erstellt. (Abb. 2.5)

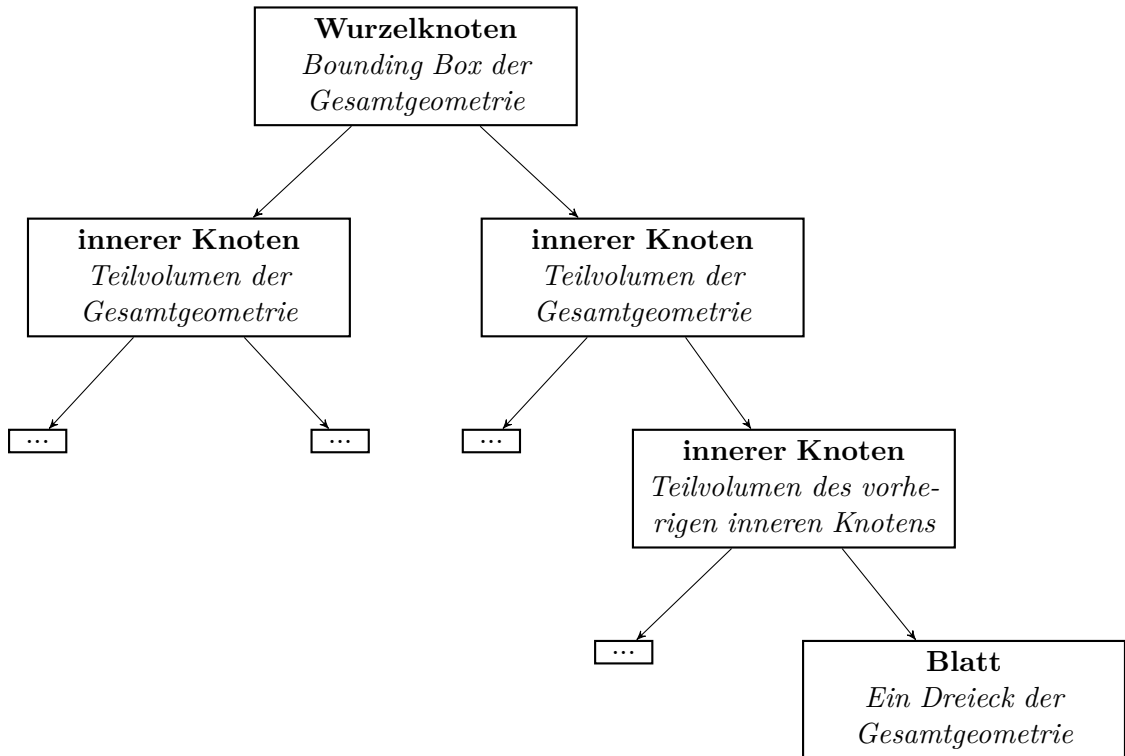


Abbildung 2.5: Baumstruktur der Bounding-Volume-Hierarchy.

Wird also von einem Strahl ein übergeordnetes Volumen nicht getroffen, so müssen auch die untergeordneten Volumina nicht behandelt werden. Der Vorteil dieses Algorithmus spiegelt sich im worst case in einem logarithmischen Aufwand wider [7]. Dies führt zu einer erheblichen Rechenzeitreduktion bzw. Performancesteigerung.

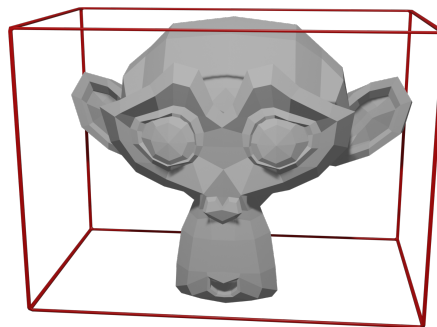


Abbildung 2.6: Bounding Box der gesamten Beispielgeometrie.

2.4 Soundmap

2.4.1 Übersicht

Um den Rechenaufwand von Lichtsimulationen in Echtzeitanwendungen wie z.B. Computerspielen zu minimieren, wird häufig auf die Lightmap zurückgegriffen. Hierbei wird mit z.B. der Raytracing-Methode bereits vor der Anwendung die Auswirkung einer fixen Lichtquelle vorsi-muliert. Die an den Geometrieoberflächen eintreffenden Lichtstrahlen erzeugen unterschiedliche Helligkeiten, die in einer Datenstruktur gespeichert werden und abschließend als Oberflächentex-tur für die Anwendung zur Verfügung stehen. Dies bietet den Vorteil, dass nun beim Abtasten der Umgebung von der Kamera aus, die Texturen bereits die Informationen der Lichtquellen enthalten.

Für die raumakustische Simulation kann nun eine “Soundmap” eingeführt werden. Diese Da-tenstruktur speichert wichtige Kenndaten einer jeden Reflexion, die während der Simulation auftritt. Diese Kenndaten eines solchen Soundmap-Eintrags sind in Tabelle 2.2 aufgelistet.

Beschreibung	Variable	Datentyp	benötigter Speicher in Byte
Vektor vom Ursprung zum Reflexionspunkt	\vec{p}_i^ξ	float(3)	12
Reflexionseinheitsvektor	\vec{r}_i^ξ	float(3)	12
Distanz von Quelle bis zur Reflexion	d_i^ξ	float	4
Produkt der Reflexionsgrade aller bis zur Refle-xion ξ beteiligten Flächen für jedes Oktavband (B)	$R_i^\xi(B)$	float(6)	24

Tabelle 2.2: Kenndaten eines Soundmap-Eintrags, entstanden durch die ξ -te Reflexion des i -ten Rays (vgl. Kapitel 2.4.2).

Das Erstellen der Soundmap ist somit von der Empfängerposition unabhängig. Zur Auswertung können nun im letzten Schritt alle Soundmap-Einträge ausgewählt werden, die bei Weiterverfol-gung des Schallstrahls den Empfänger schneiden. Dieser Schritt erfordert verhältnismäßig wenig Rechenaufwand und kann nach Veränderung der Empfängerposition schnell (einige Sekunden) wiederholt werden. Als Nachteil ist jedoch ein hoher Speicheraufwand zu erwähnen (52 Byte pro Eintrag). So generiert z.B. eine Simulation mit 10^6 Rays und 10 Reflexionen pro Ray bereits eine Soundmap von der Größe $10^6 \cdot 10 \cdot 52 \text{ Byte} = 520 \text{ MByte}$.



Abbildung 2.7: Grafische Darstellung der Soundmap in einem fiktiven Konzertsaal. Die Positionen der sichtbaren Punkte entsprechen \vec{p}_i^ξ . Die Farbgebung wird zwischen 1 (rot) und 0 (blau) linear interpoliert und ist vom Parameter $R_i^\xi(B)$ abhängig.

2.4.2 Berechnung der Parameter eines Soundmap-Eintrags

Ein Soundmap-Eintrag kann immer dem i -ten Ray und dessen ξ -ter Reflexion zugeordnet werden. Die Parameter aus Tabelle 2.2 lassen sich wie folgt rekursiv aus den Informationen der vorherigen Reflexion berechnen.

Vektor vom Ursprung zum Reflexionspunkt

$$\vec{p}_i^0 = \vec{p}_{\text{Source}}$$

$$\vec{r}_i^0 = \begin{pmatrix} \cos \theta_i \sin \phi_i \\ \sin \theta_i \sin \phi_i \\ \cos \phi_i \end{pmatrix} \quad (\text{siehe Gleichung 2.3})$$

$$\vec{p}_i^\xi = \vec{p}_i^{\xi-1} + d_i^{\xi, \xi-1} \cdot \vec{r}_i^{\xi-1} \quad (2.4)$$

$d_i^{\xi, \xi-1}$... Distanz $\|\vec{p}_i^\xi - \vec{p}_i^{\xi-1}\|$, bekannt durch Schnittpunktkalkulation mit Embree (Kapitel 2.3)

Reflexionseinheitsvektor

$$\vec{r}_i^\xi = 2 \cdot \langle -\vec{r}_i^{\xi-1}, \vec{n}_{\text{Face}} \rangle \cdot \vec{n}_{\text{Face}} + \vec{r}_i^{\xi-1} \quad (2.5)$$

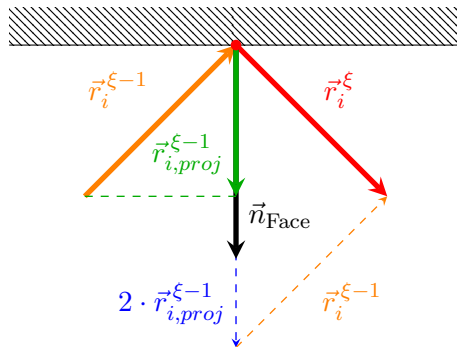


Abbildung 2.8: Ermittlung des Reflexionseinheitsvektors \vec{r}_i^ξ mit $\vec{r}_{i,proj}^{\xi-1} = \langle -\vec{r}_i^{\xi-1}, \vec{n}_{\text{Face}} \rangle \cdot \vec{n}_{\text{Face}}$.

Distanz von Quelle bis zur Reflexion

$$d_i^0 = 0$$

$$d_i^\xi = d_i^{\xi-1} + d_i^{\xi, \xi-1} \quad (2.6)$$

Reflexionsgrad

$$R_i^0(B) = 1$$

$$R_i^\xi(B) = R_i^{\xi-1}(B) \cdot (1 - \alpha_{\text{Face}}(B)) \quad (2.7)$$

$\alpha_{\text{Face}}(B)$... oktavbandabhängiger Absorptionsgrad

2.4.3 Auswertung an der Empfängerposition

Zur Auswertung sollen nun alle Soundmap-Einträge ausgewählt werden, die bei Weiterverfolgung des Schallstrahls den Empfänger schneiden.

Empfängerformat

Um auf einen Schnittpunkt (Schallstrahl mit Empfänger) schließen zu können, muss zuallererst die Form und Größe des Empfängers definiert werden. Da der an der Empfängerposition ankommende Schallenergie direkt mit der Größe des Empfängers in Verbindung steht, muss gerade dieser Parameter für akkurate Ergebnisse sorgfältig gewählt werden. In der Literatur werden für diesen Prozess einige Varianten beschrieben. In einer Veröffentlichung von Xiangyang et al. werden einige Ansätze miteinander verglichen. Zusätzlich wird eine andere Herangehensweise vorgestellt, welche sich als äußerst treffend erweist [8]. Es wird dabei von einer Kugel mit einem variablen Radius (r_{Receiver}), welcher sich aus dem Raumvolumen (V_{Room}), der Distanz zwischen Quelle und Empfänger (d_{SR}) und der Anzahl an Strahlen (N) zusammensetzt, ausgegangen:

$$r_{\text{Receiver}} = k \cdot d_{SR} \cdot \sqrt{\frac{4}{N}}$$

$$k = \log_{10}(V_{\text{Room}}) \quad (2.8)$$

Vorauswahl

Um eine effiziente Vorauswahl zu treffen, kann man folgende Überlegungen anstellen (Abb. 2.9): Ein Ray schneidet den Empfänger sicher, wenn der Winkel θ so klein ist, dass die Distanz $\|\vec{CA}\|$ kleiner als der Radius der Empfängerkugel ist. \vec{CA} kann durch folgende Berechnungen ermittelt werden:

$$\tan \theta = \frac{\|\vec{CA}\|}{\|\vec{BC}\|}$$

$$\|\vec{CA}\| = \tan \theta \cdot \|\vec{BC}\| \quad (2.9)$$

Der Winkel θ lässt sich aus der Definition des Skalarprodukts berechnen.

$$\begin{aligned} \langle \vec{BC}, \vec{r}_i^\xi \rangle &= \|\vec{BC}\| \cdot \underbrace{\|\vec{r}_i^\xi\|}_{=1} \cdot \cos(\theta) \\ \theta &= \arccos\left(\frac{\langle \vec{BC}, \vec{r}_i^\xi \rangle}{\|\vec{BC}\|}\right) \quad \text{Subst.: } x = \frac{\langle \vec{BC}, \vec{r}_i^\xi \rangle}{\|\vec{BC}\|} \\ \|\vec{CA}\| &= \tan(\arccos(x)) \cdot \|\vec{BC}\| \\ \|\vec{CA}\| &= \frac{\sqrt{1-x^2}}{x} \cdot \|\vec{BC}\| \\ \|\vec{CA}\| &= \frac{\sqrt{1 - \left(\frac{\langle \vec{BC}, \vec{r}_i^\xi \rangle}{\|\vec{BC}\|}\right)^2}}{\left(\frac{\langle \vec{BC}, \vec{r}_i^\xi \rangle}{\|\vec{BC}\|}\right)} \cdot \|\vec{BC}\| \end{aligned} \quad (2.10)$$

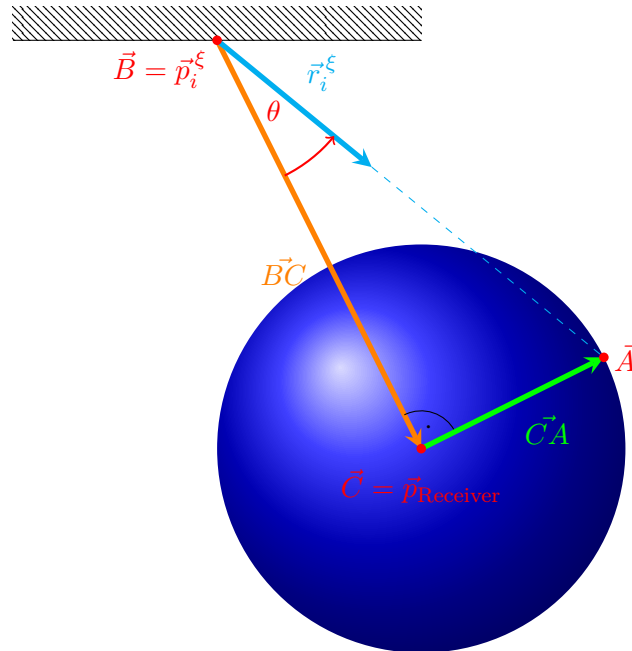


Abbildung 2.9: Schnelle Vorauswahl der Punkte aus der Soundmap durch Berechnung von \vec{CA} .

Durch geschicktes Umformen sind nur wenige einfache Rechenoperationen notwendig, die vom Prozessor sehr schnell ausgeführt werden können. Weiters konnte der Prozess durch Verwendung von SIMD-Operationen (Single Instruction Multiple Data) [9] um ein vielfaches beschleunigt werden. Wie in Abbildung 2.9 zu erkennen ist, werden teilweise Rays als ungültig eingestuft, obwohl sie den Empfänger schneiden. Diese Ungenauigkeit wird allerdings aufgrund der deutlichen Performancesteigerung in Kauf genommen.

Nach der Vorauswahl der geeigneten Soundmap-Einträge, muss im nächsten Schritt noch von jedem gültigen Eintrag mittels Embree (Kapitel 2.3) ein Ray von \vec{p}_i^ξ nach $\vec{p}_{\text{Receiver}}$ geschickt werden, um Geometrien (zum Beispiel das eingebrachte Mobiliar) zwischen Reflexionspunkt und Empfänger zu erkennen. Falls der Empfänger verdeckt wird, muss der Eintrag nicht berücksichtigt werden.

2.4.4 Erstellen der oktavbandabhängigen Impulsantworten

Jeder von der Kugelquelle ausgehende Strahl mit Index i ist mit einer Energie von 1 J gewichtet. Die vorausgewählten Soundmapeninträge aus Kapitel 2.4.3 stammen jeweils von einem Schallstrahl i und repräsentieren dessen ξ -ten Reflexionspunkt ($1 \leq \xi \leq \text{lifespan}$).

Da die Verlängerung des Reflexionseinheitsvektor \vec{r}_i^ξ den Empfänger schneidet, entspricht die detektierte Energie am Empfänger dem Soundmapeintrag $R_i^\xi(B)$. Diese kann als gewichtete Dirac-Distribution modelliert werden.

$$R_i^\xi(B) \cdot \delta_i^\xi(t - t_i^{\text{Receiver}}) = \begin{cases} R_i^\xi(B) & \text{für } t = t_i^{\text{Receiver}} \\ 0 & \text{sonst} \end{cases} \quad (2.11)$$

t_i^{Receiver} ist hierbei die Dauer, welche der jeweilige Schallstrahl benötigt um beim Empfänger einzutreffen. Diese berechnet sich aus der gesamten Weglänge des Rays i zwischen Quelle und Empfänger über alle Reflexionspunkte ($d_{i,\text{total}}$).

$$d_{i,\text{total}} = d_i^\xi + \|\vec{p}_{\text{Receiver}} - \vec{p}_i^\xi\|$$

$$t_i^{\text{Receiver}} = \frac{d_{i,\text{total}}}{c} \quad (2.12)$$

Die Impulsantwort soll mit einer Abtastrate von f_s diskretisiert werden. Dazu wird aus der kontinuierlichen Zeit t_i^{Receiver} die diskrete Samplenummer n_i^{Receiver} ermittelt.

$$F(t) \longrightarrow F[n] = F(nT) = F\left(\frac{n}{f_s}\right)$$

$$n_i^{\text{Receiver}} = \lfloor t_i^{\text{Receiver}} \cdot f_s \rfloor \quad \{n_i^{\text{Receiver}} \in \mathbb{N} \mid 0 \leq n_i^{\text{Receiver}} < N\} \quad (2.13)$$

N entspricht der Anzahl der Samples in der Impulsantwort. Alle Energien der eintreffenden Strahlen werden nun zur oktavbandabhängigen Energieimpulsantwort $h_{E,B}$ aufaddiert.

$$h_{E,B}[n] = \sum_{\Gamma} R_i^\xi(B) \cdot \delta_i^\xi[n - n_i^{\text{Receiver}}] \quad (2.14)$$

Γ ... Menge aller aus der Soundmap nach Kapitel 2.4.3 ausgewählten Einträge

Da der Direktschall nicht Teil der Soundmap ist, muss dieser Energieanteil noch zusätzlich berechnet werden. Hierfür wird zuerst mittels Embree ein Strahl von der Quelle in Richtung Empfänger geschickt um sicherzustellen, dass keine Geometrie den Empfänger verdeckt. Falls er verdeckt wird, werden keine weiteren Berechnungen angestellt. Ansonsten werden die einfallenden Direktstrahlen über die Strahlendichte ρ auf der Quellsphärenfläche S_H mit dem Radius $d_{S,R}$ abgeschätzt.

$$d_{S,R} = \|\vec{p}_{\text{Receiver}} - \vec{p}_{\text{Source}}\|$$

$$S_H = 4\pi(d_{S,R})^2$$

$$\rho = \frac{\#\text{Rays}}{S_H} \quad (2.15)$$

Auf die Querschnittsfläche des Empfängers bezogen, ergibt sich die Anzahl der Strahlen des Direktschalls.

$$\#\text{Rays}_{direct} = \pi \cdot (r_{Receiver})^2 \cdot \rho \quad (2.16)$$

Die Direktschallstrahlen haben keine Energie durch Reflexionen verloren, sodass ein Strahl eine Energie von 1 J trägt. Somit beläuft sich die Energie des Direktschalls auf $\#\text{Rays}_{direct}$. Ähnlich wie in Gleichung 2.13 kann die diskrete Samplenummer n_{direct} bestimmt werden, bei welcher der Direktschall beim Empfänger eintrifft.

$$\begin{aligned} n_{direct} &= \lfloor \frac{d_{S,R}}{c} \cdot f_s \rfloor \\ h_{E,B}[n_{direct}] &= \#\text{Rays}_{direct} \end{aligned} \quad (2.17)$$

Die Luftabsorption kann nun berücksichtigt werden, indem jeder Wert der Energieimpulsantwort mit einer Exponentialfunktion multipliziert wird [2]. Um von der Energieimpulsantwort auf eine Schalldruckimpulsantwort $h_B[n]$ zu kommen, muss zusätzlich aus jedem Wert die Wurzel gezogen werden ($E \sim p^2$).

$$\begin{aligned} d_n &= c \cdot \frac{n}{f_s} \\ h_B[n] &= \sqrt{h_{E,B}[n] \cdot \exp(-2 \cdot \mu(B) \cdot d_n)} \\ &= \sqrt{h_{E,B}[n] \cdot \exp\left(-2 \cdot 74 \cdot 10^{-9} \frac{(f_{m,B})^2}{\varphi} \cdot d_n\right)} \end{aligned} \quad (2.18)$$

d_n ... zurückgelegter Weg der an der n-ten Samplenummer eingetroffenen Reflexionen

Die nun in $h_B[n]$ enthaltenen Werte sind durch die Annahme entstanden, dass die Energie eines Schallstrahls 1 J sei. Somit stehen sie in keinem Kontext zur eingestellten Schallleistung der Quelle. Um dem entgegenzuwirken, kann der Schalldruck des Direktschalls am Empfänger über die eingestellte Schallleistung der Quelle berechnet werden.

$$\begin{aligned} I_{direct} &= \frac{P_{Source}}{4\pi(d_{S,R})^2} = \frac{(p_{direct}(B))^2}{Z_0} \\ \Rightarrow p_{direct}(B) &= \sqrt{\frac{P_{Source} \cdot \exp(-2 \cdot \mu(B) \cdot d_{S,R}) \cdot Z_0}{4 \cdot \pi \cdot (d_{S,R})^2}} \end{aligned} \quad (2.19)$$

Infolgedessen kann die Impulsantwort zuerst normiert und dann zum Wert des Schalldrucks $p_{direct}(B)$ in Bezug gesetzt werden.

$$h_{B,cal}[n] = \frac{h_B[n]}{h_B[n_{direct}]} \cdot p_{direct}(B) \quad (2.20)$$

$h_{B,cal}[n]$... auf den Direktschall $p_{direct}(B)$ kalibrierte Impulsantworten $h_B[n]$

Die in $h_{B,cal}[n]$ enthaltenen Werte entsprechen nun den durch die Schallleistung der Quelle ausgelösten Schalldrücken. Dieser Rechenschritt ist allerdings in ARTWA nicht implementiert, da die Impulsantwort beim Export als .wav-Datei auf den Wertebereich der 16 bit-Quantisierung

normalisiert wird und somit die Information über den Wertebereich des Schalldrucks verloren geht. Falls kein Direktschall ermittelt wurde, da der Empfänger verdeckt war, muss für diesen Schritt trotzdem ein $p_{direct}(B)$ ermittelt werden. Es wird dafür kein Hindernis zwischen Quelle und Empfänger angenommen.

2.4.5 Erstellen der breitbandigen Impulsantwort

Da die oktavbandabhängigen Impulsantworten $h_B[n]$ breitbandige Dirac-Impulse enthalten, müssen sie mithilfe von entsprechenden Bandpassfiltern gefiltert werden. Hierfür durchlaufen sie den jeweiligen Bandpassfilter $g_B[n]$ mit Mittenfrequenz $f_{m,B}$ einer Filterbank (Abb. 2.10). Als Filter wird eine IIR-Implementation von Besselfiltern der 4. Ordnung verwendet [10]. Die breitbandige Impulsantwort $h[n]$ ergibt sich aus der Summe aller gefilterten $h_B[n]$:

$$h[n] = \sum_B h_B[n] * g_B[n] \quad (2.21)$$

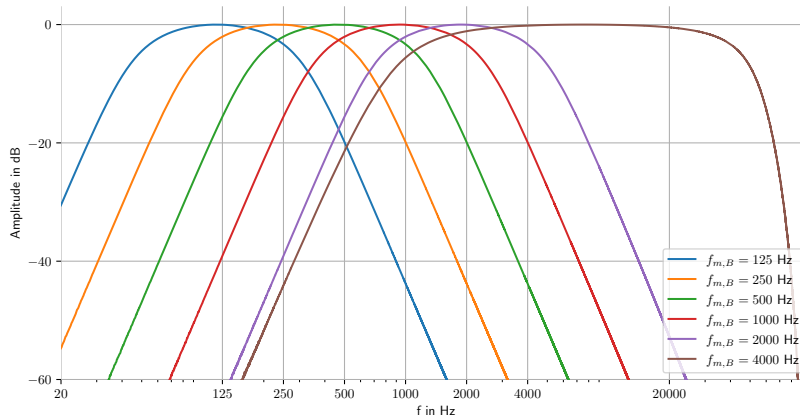


Abbildung 2.10: Amplitudenspektrum der Filterbank bestehend aus Bandpassfiltern $g_B[n]$ mit der jeweiligen Mittenfrequenz $f_{m,B}$ zur Zerlegung der Impulsantwort in die Oktavbänder.

Erwähnenswert ist die Beschaffenheit des Bandpassfilters für $f_{m,B} = 4000$ Hz. Die untere Grenzfrequenz $f_{u,B}$ beträgt $f_{m,B}/\sqrt{2} = 2828$ Hz, die obere Grenzfrequenz liegt bei $f_{o,B} = 20$ kHz. Daraus ergibt sich eine Bandbreite von $f_{o,B} - f_{u,B} = 17172$ Hz und eine tatsächliche Mittenfrequenz des Bandpasses von $\sqrt{f_{u,B} \cdot f_{o,B}} = 7521$ Hz. Diese Wahl ermöglicht bei Faltung mit einer Beispielaudiodatei eine Abbildung des gesamten hörbaren Frequenzbereichs.

3

Leitfaden zur ARTWA Software

3.1 Vorkonfiguration

Vor der eigentlichen Raumsimulation ist es notwendig, einen Raum mit für ARTWA geeigneten Spezifikationen zu erstellen. Um diesen Prozess zu erleichtern und einen fehlerfreien Ablauf zu gewährleisten, wird ein Addon für Blender zur Verfügung gestellt. Blender ist eine Open Source Software, welche nach dem kostenfreien Download auf dem Simulationsrechner installiert werden muss. (mind. v2.8.x)

3.1.1 Addon für Blender

Zur Installation des Addons muss die mitgelieferte Datei `addon_install.exe` ausgeführt werden. Blender öffnet sich dabei automatisch und das Addon "Room Simulation" kann in der Sidebar des 3D Viewports eingesehen werden. Falls diese nicht sichtbar ist, kann sie unter dem Menüpunkt `View` ▶ `Sidebar` (Tastenkürzel `N`) aufgerufen werden.

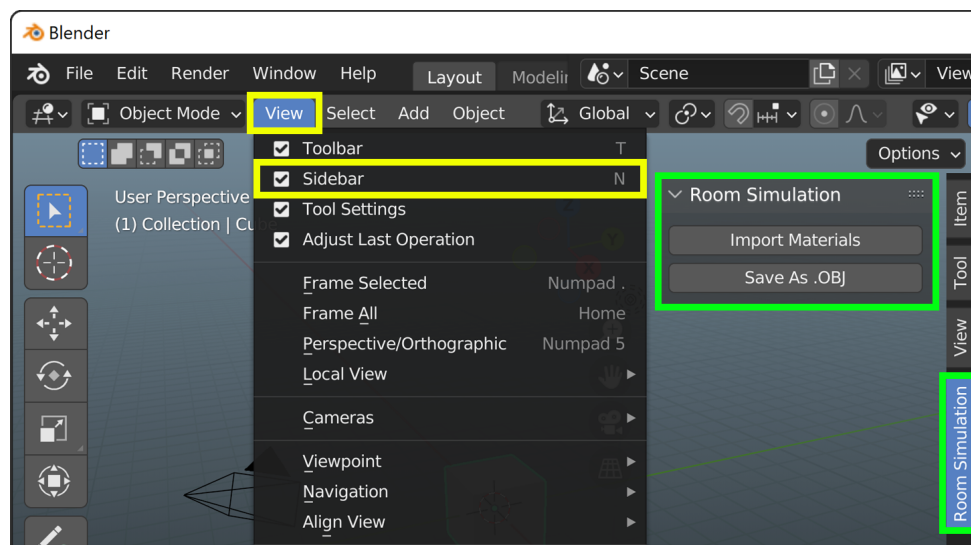


Abbildung 3.1: Navigation zum Aktivieren der Sidebar innerhalb von Blender (gelb) und das entwickelte Addon "Room Simulation" (grün).

Folgende Funktionen stehen zur Auswahl:

- **Import Materials**

Es öffnet sich ein Fenster des Dateibrowsers, in welchem nach einer .csv-Datei gesucht werden kann. Dort kann der Anwender die Materialien mit deren Absorptionseigenschaften in folgendem Format hinterlegen:

Name	Absorptionsgrade der Oktavbänder in Hz						Farbe		
	125	250	500	1000	2000	4000	R	G	B
Beton	0.01	0.01	0.01	0.02	0.02	0.02	0.34	0.34	0.34
Marmor	0.01	0.01	0.02	0.02	0.03	0.03	0.90	0.87	0.78
Fensterglas	0.35	0.25	0.18	0.12	0.07	0.04	0.52	0.57	0.75
⋮	...								

Tabelle 3.1: Materialliste mit Beispielmaterialien als .csv-Datei.

- **Save As .OBJ**

Über den blenderinternen Dateibrowser kann ein Speicherort und -name für den .obj-Datei-Export festgelegt werden. Die für die Raumsimulation wesentlichen Parameter (vgl. Kap. 2.2) werden hierbei automatisch gesetzt.

3.1.2 Anwendung mit einem beliebigen CAD-Programm

Es wird dringlich empfohlen, die Raumerstellung mittels Blender durchzuführen. Wurde in einem anderen CAD-Programm bereits ein Raum erstellt, so kann dieser als .obj-Datei exportiert und in Blender importiert werden. Anschließend können nun über das beschriebene Addon Materialien geladen und den Flächen zugewiesen werden.

3.1.3 Zuweisung von Materialien

Das Zuweisen von den bereits geladenen Materialien kann nun, wie in Blender üblich, erfolgen. Da diese Vorgehensweise versionsabhängig sein kann, wird an dieser Stelle auf die Dokumentation von Blender verwiesen [11].

3.2 ARTWA

Nachdem die Vorkonfiguration (Kap. 3.1) erfolgreich abgeschlossen wurde, kann mit der eigentlichen Simulation begonnen werden. Um das Programm ARTWA zu starten, wird im Hauptverzeichnis eine “ARTWA.exe” bereitgestellt. Es öffnet sich ein Fenster mit einem leeren 3D Viewport und einer Einstellungssidebar an der linken Seite.

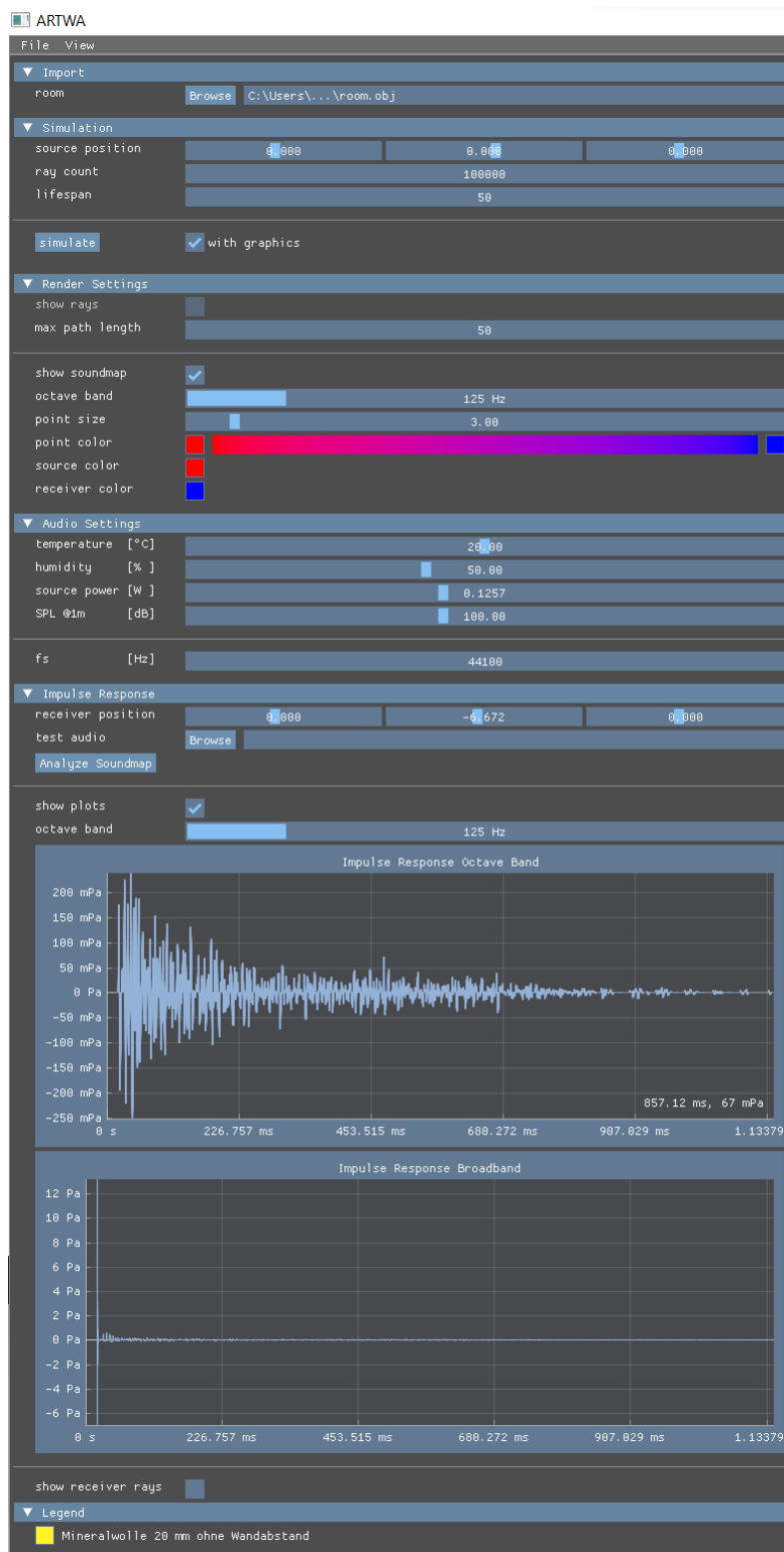


Abbildung 3.2: Einstellungssidebar innerhalb des Grafischen User Interface der ARTWA Software.

3.2.1 Import

Auf dieser Sidebar kann auch ein “browse”-Button gefunden werden, welcher die Möglichkeit für den Import des zuvor erstellten Raummodells bietet. Als Format wird ausschließlich eine .obj-Datei akzeptiert. Die gleichnamige .mtl-Datei, welche die Materialspezifikationen beinhaltet, muss dabei im selben Verzeichnis vorliegen (wurde automatisch vom Addon erstellt). Nach dem Import ist das Koordinatensystem und somit auch der Ursprung in ARTWA identisch mit dem in der .obj-Datei verwendeten. Sollten beim Import des Raums Probleme auftreten, können dem Kap. 2.2 mögliche Fehlerquellen entnommen werden.



Abbildung 3.3: Abschnitt “Import” der Einstellungssidebar.

Nachdem ein Raummodell ausgewählt wurde, wird dieses im 3D-Viewport sichtbar:

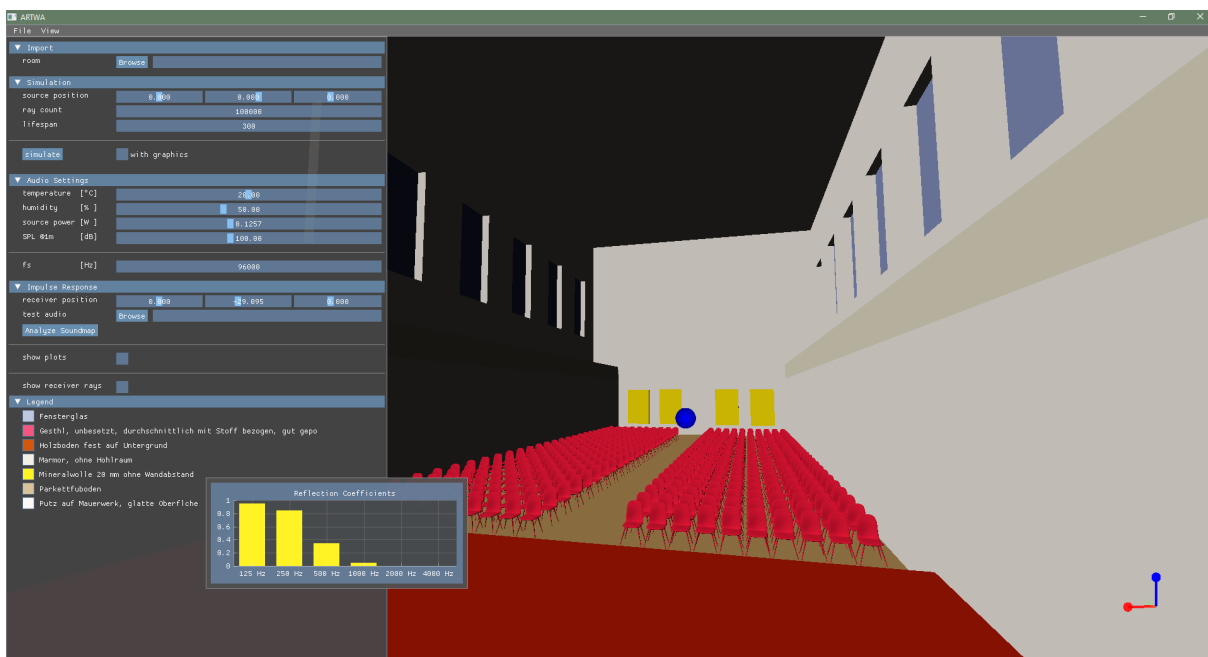


Abbildung 3.4: ARTWA direkt nachdem ein Raummodell importiert wurde.

3.2.2 Simulation

Die nachfolgenden Parameter müssen vor der Simulation gesetzt werden. Eine Veränderung danach wird erst mit einem erneuten Aufruf des “simulate”-Buttons übernommen.

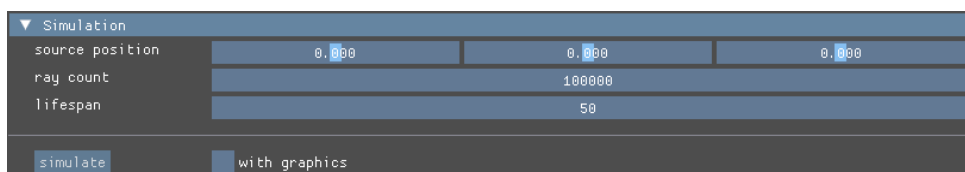


Abbildung 3.5: Abschnitt “Simulation” der Einstellungssidebar.

- **source position**
Die Schallquellenposition kann mit den drei angezeigten Schieberegler entlang der Raumachsen (x, y, z) verschoben werden. Der Wertebereich wurde hierbei über die Bounding Box (Abb. 2.6) der Gesamtgeometrie begrenzt.
- **ray count**
Dieser Parameter gibt an, wie viele Schallstrahlen von der Schallquelle ausgesendet werden sollen (Kap. 2.1). Je höher die eingegebene Anzahl, desto akkurater die Simulation.
- **lifespan**
Mit der lifespan wird die maximale Reflexionsanzahl eines Schallstrahls bestimmt. Dieser Wert darf nicht zu klein gewählt werden, da ansonsten die Impulsantwort frühzeitig abgeschnitten wird.
- **simulate**
Dieser Button startet die Simulation und generiert die Strahlen bzw. Soundmap.
- **with graphics**
Über diese Checkbox kann bestimmt werden, ob die Ergebnisse der Simulation im 3D-Viewport grafisch dargestellt werden. Diese Operation kann sich, je nach ray count bzw. lifespan, mitunter als sehr zeitaufwendig erweisen (bis zu 1 Minute).
Zusätzlich wird bestimmt, ob der Abschnitt Render Settings (Kapitel 3.2.3) im GUI sichtbar wird oder nicht. Direkt nach dem Start von ARTWA ist der Haken der Checkbox standardmäßig nicht gesetzt.

3.2.3 Render Settings

Mit diesem Abschnitt kann die Visualisierung beeinflusst werden. Die bei der Simulation erstellten Daten bleiben hierbei immer vollständig im Hintergrund erhalten, lediglich die grafische Darstellung ändert sich.

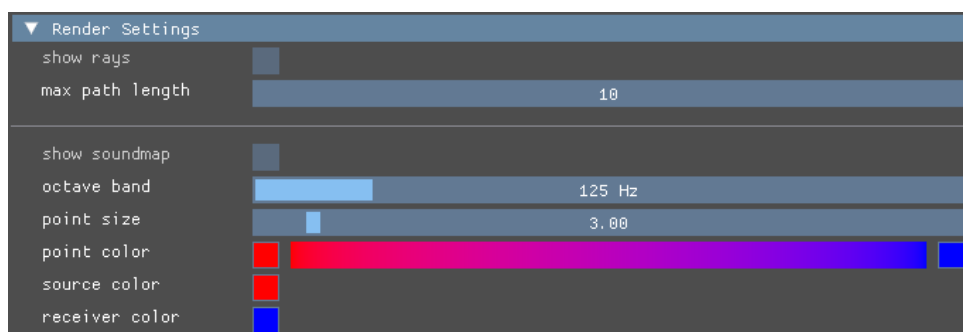


Abbildung 3.6: Abschnitt “Render Settings” der Einstellungssidebar.

- **show rays**
Diese Checkbox kann erst nach einer abgeschlossenen Simulation gesetzt werden. Zusätzlich können Schallstrahlen nur bis zu einem ray count von 4000 grafisch angezeigt werden.
- **max path length**
Dieser Parameter gibt an, bis zu welcher Reflexionsordnung der Schallstrahl gezeichnet werden soll. Der Wertebereich ist definiert als: $0 \leq \text{max path length} \leq \text{lifespan}$

- **show soundmap**

Bei Aktivierung lassen sich auf den Geometrieflächen die Schallreflexionen in Form von farbigen Punkten darstellen (Abb. 2.7). Die Einfärbung entspricht dabei der Energie des Schallstrahls unmittelbar nach der jeweiligen Reflexion.

- **octave band**

Da Absorptionsgrade üblicherweise in Oktavbändern angegeben werden, kann auch für jedes einzelne Band die jeweilige Soundmap abgefragt werden. Der Schieberegler bestimmt hierbei das aktive Band.

- **point size**

Die Punktgröße der Soundmap-Einträge kann über diesen Wert stufenlos verändert werden.

- **point color**

Die Punktfarben der Soundmap können über diese Zeile definiert werden. Während die linke der beiden Farben (Abb. 3.6, rot) die Ursprungsleistung eines einzelnen Schallstrahls ($\hat{=} \frac{\text{source power}}{\text{ray count}}$) widerspiegelt, so nimmt diese linear zur rechten Farbe hin ab (Abb. 3.6, blau). Mit einem Mausklick auf eines der beiden Farbfelder, kann über einen Farbmischer eine eigene Farbe gewählt werden.

- **source color**

Die im 3D Viewport farbliche Repräsentation der Quelle kann über diesen Farbwähler verändert werden.

- **receiver color**

Die im 3D Viewport farbliche Repräsentation des Empfängers kann über diesen Farbwähler verändert werden.

3.2.4 Audio Settings

In dieser Einstellungssektion können physikalische Gegebenheiten des Raums bzw. der Quelle festgelegt werden. Weiters kann die Abtastfrequenz der Impulsantwort eingesehen werden.

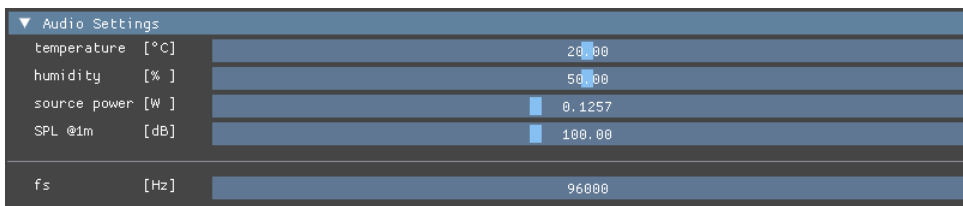


Abbildung 3.7: Abschnitt "Audio Settings" der Einstellungssidebar.

- **temperature**

Ein wesentlicher Parameter in der Akustik ist die Lufttemperatur ϑ . Von ihr hängen die Luftdichte ρ , die Schallgeschwindigkeit c und die Schallkennimpedanz Z_0 ab. Es kann eine Temperatur von 0 – 40°C gewählt werden. Bei Änderung des Temperaturwertes werden alle abhängenden Parameter nach folgenden Formeln aktualisiert [2].

$$\rho = \frac{\rho_0}{1 + \alpha \cdot \vartheta} \quad (3.1)$$

$$\alpha = \frac{1}{273,15} \quad \dots \text{Temperaturkoeffizient } [^{\circ}\text{C}^{-1}]$$

$$\rho_0 = 1,29 \quad \dots \text{Luftdichte bei } \vartheta = 0^\circ\text{C} \left[\frac{\text{kg}}{\text{m}^3} \right]$$

$$c = c_0 \cdot \left(1 + \frac{\alpha \cdot \vartheta}{2} \right) \quad (3.2)$$

$$c_0 = 331,6 \quad \dots \text{Schallgeschwindigkeit bei } \vartheta = 0^\circ\text{C} \left[\frac{\text{m}}{\text{s}} \right]$$

$$Z_0 = \rho \cdot c \quad (3.3)$$

- **humidity**

Um den Effekt der Luftdissipation simulieren zu können, ist die Angabe der Luftfeuchtigkeit in % nötig. Anwendung findet der Parameter in Gleichung 2.18. Die Gleichung ist eine Näherung und gilt nur für $30\% \leq \varphi \leq 80\%$, sodass auch nur dieser Wertebereich ausgewählt werden kann.

- **source power**

Der User hat die Möglichkeit, die emittierte Schalleistung der Quelle $P_{Source}[W]$ anzugeben. Dieser Wert hängt direkt mit dem nachfolgenden Parameter "SPL @1m" zusammen. Der Wertebereich erstreckt sich zwischen $126 \mu\text{W}$ (SPL @1m = 70 dB) und $1,256 \text{ kW}$ (SPL @1m = 140 dB), die Umrechnung in SPL @1m ist möglich durch:

$$\text{SPL @1m} = 10 \cdot \log_{10} \left(\frac{P_{Source}}{4\pi \cdot 10^{-12} \text{ W}} \right) \quad (3.4)$$

- **SPL @1m**

Dieser Wert wird oft im Datenblatt von Lautsprechern angegeben und ist daher eine einfache Möglichkeit, die Schalleistung indirekt einzugeben. Die Grenzen sind $70 - 140 \text{ dB}_{SPL}$, eine Umrechnung in P_{Source} erfolgt durch:

$$P_{Source} = 10^{\text{SPL @1m}/10} \cdot 4\pi \cdot 10^{-12} \text{ W} \quad (3.5)$$

- **fs**

Die Abtastfrequenz der Impulsantwort ist von der jeweilig geladenen "test audio"-Datei (Kapitel 3.2.5) abhängig und kann in diesem Feld abgelesen, jedoch nicht editiert werden.

3.2.5 Impulse Response

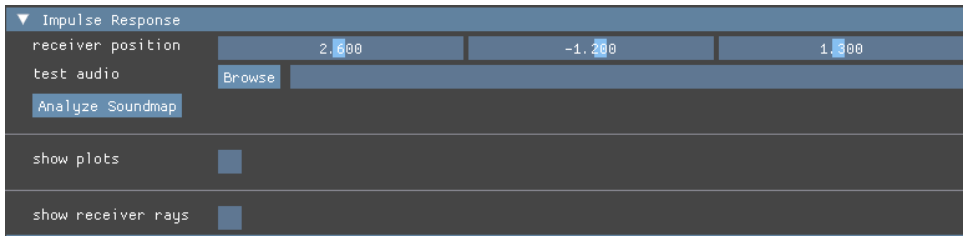


Abbildung 3.8: Abschnitt “Impulse Response” der Einstellungssidebar.

- **receiver position**

Die Empfängerposition kann mit den drei angezeigten Schieberegler entlang der Raumachsen (x, y, z) verschoben werden. Der Wertebereich wurde hierbei über die Bounding Box (Abb. 2.6) der Gesamtgeometrie begrenzt.

- **test audio**

Über den “Browse”-Button kann an dieser Stelle die jeweilige Beispiel-Audiodatei in Form einer “.wav-Datei” geladen werden. Die Abtastfrequenz der Impulsantwort wird dabei automatisch auf jene der geladenen Datei festgelegt.

- **Analyze Soundmap**

Über diesen Button wird an der mit “receiver position” festgelegten Position im Raum die Soundmap analysiert um anschließend eine Impulsantwort zu erhalten. Sollte diese auch mit einer test audio Datei gefaltet werden, so muss im Punkt “test audio” eine “.wav-Datei” importiert werden. Nach erneutem drücken des Buttons, wird das Ergebnis der Faltung automatisch über den “Default Audio Treiber” des Systems abgespielt.

- **show plots**

Diese Checkbox bestimmt, ob die Plots aus Abbildung 3.9 angezeigt werden sollen. Es handelt sich dabei um interaktive Plots, die sich beliebig verschieben und vergrößern lassen. Bei längeren Impulsantworten kann es hier zum Einbruch der Bildwiederholungsrate von ARTWA kommen. In diesem Fall empfiehlt sich ein Ausblenden der Plots. Der Schalldruck ist, wie in Kapitel 2.4.4 beschrieben, nicht auf die eingestellte Schallleistung der Quelle kalibriert.

1. **octave band**

Mit diesem Schieberegler können die verschiedenen Impulsantworten der einzelnen Oktavbänder abgefragt werden. Die Daten werden anschließend im Plot “Impulse Response Octave Band” angezeigt.

2. **Impulse Response Octave Band**

In diesem Plot wird die Impulsantwort eines einzelnen Oktavbandes dargestellt.

3. **Impulse Response Broadband**

Die breitbandige Impulsantwort (Kapitel 2.4.5) wird in diesem Fenster grafisch dargestellt.

- **show receiver rays**

Mit dieser Checkbox wird die grafische Darstellung jener Strahlen beeinflusst, welche bei der Soundmap-Auswertung (Kapitel 2.4.3) berücksichtigt werden (pink). Zusätzlich werden in gelb Strahlen angezeigt, welche den Empfänger geschnitten hätten, allerdings von einem Gegenstand verdeckt wurden.

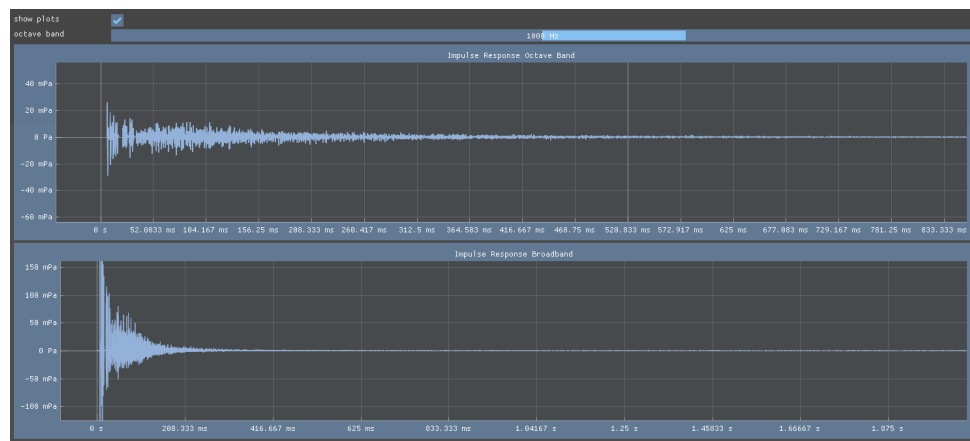


Abbildung 3.9: Plots der gefilterten Impulsantwort eines Bandes (oben) und der breitbandigen Impulsantwort (unten).

3.2.6 Materiallegende

Nachdem ein Raum in den 3D Viewport geladen wurde, wird in der Sidebar eine Materiallegende sichtbar (Abb. 3.10). Diese bringt den Materialnamen mit der jeweiligen Farbe in Verbindung. Zusätzlich kann mit der Maus durch darübergleiten (hovern) Auskunft über die Reflexionsgrade der einzelnen Oberflächen gegeben werden.

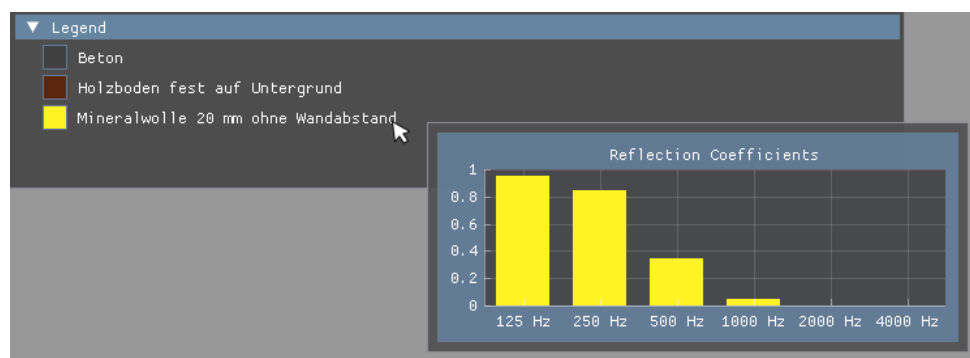


Abbildung 3.10: Abschnitt "Legend" der Einstellungssidebar.

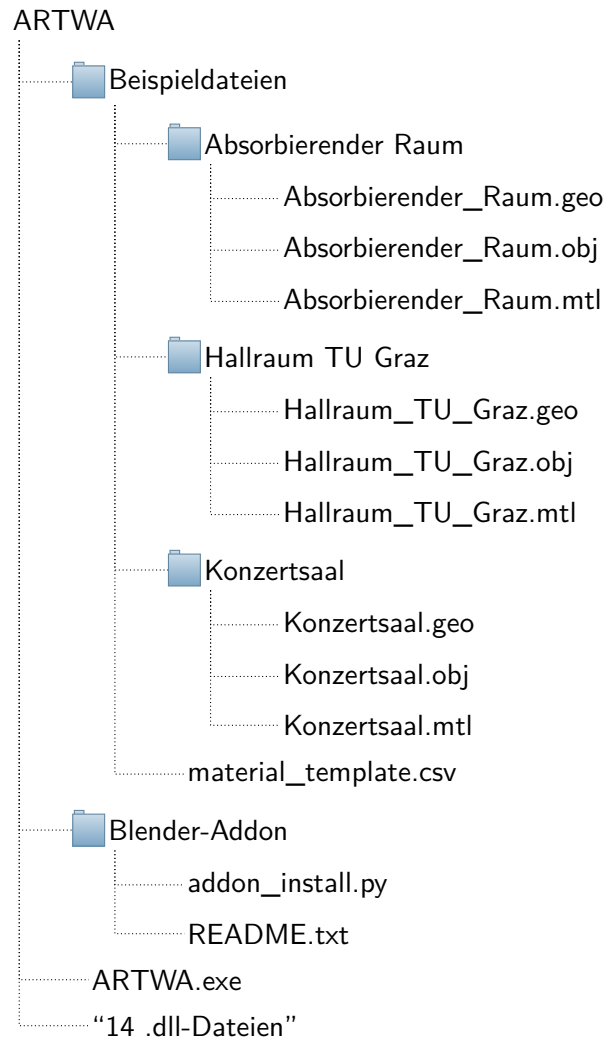


Abbildung 3.11: Alle bei ARTWA mitgelieferten Dateien.

Um die Raummodellierung, wie in Kapitel 3.1 beschrieben, in Blender durchführen zu können, muss Blender unter folgendem Link heruntergeladen und installiert werden:

<https://www.blender.org/download/>

4

Vergleich mit kommerzieller Software

Um die Ergebnisse von ARTWA innerhalb eines wissenschaftlichen Kontextes diskutieren zu können, wird an dieser Stelle ein komparables, kommerzielles Softwareprodukt als Vergleichsobjekt herangezogen. Zusätzlich wird im ersten Vergleichsszenario ein Ergebnis von einer realen Messumgebung jenen der Simulation gegenüber gestellt. Das Messergebnis wurde im Rahmen der Masterarbeit “Analyse des modalen Schallfeldes zur Untersuchung der Funktionsweise von Kantenabsorbern” als 4. Messreihe (Lautsprecherposition 1, Mikrofonposition 4) angefertigt. [12]

4.1 Hallraum des Labors für Bauphysik an der TU Graz

4.1.1 Raumeigenschaften

Der Raum besteht aus einem Quader mit der Grundfläche von 8.34 m x 5.99 m und der Höhe 4.90 m. Daraus ergibt sich ein Volumen von 245 m³. Mithilfe der ITA-Toolbox [13] wurden die oktavbandabhängigen Nachhallzeiten (T30) ermittelt. In Anlehnung an EN ISO 354 [14] wurden im nächsten Schritt mit Formel 4.1 die Absorptionseigenschaften der Wände des Hallraums bestimmt:

$$\alpha(B) = \frac{24 \cdot \ln(10) \cdot V}{c \cdot S \cdot T(B)} \quad (4.1)$$

Materialname	Oberfläche S	α_{125}	α_{250}	α_{500}	α_{1000}	α_{2000}	α_{4000}
Beton	240 m ²	0.007	0.011	0.013	0.017	0.027	0.056

Tabelle 4.1: Absorptionsgrade des Wandmaterials vom Hallraum des Labors für Bauphysik.

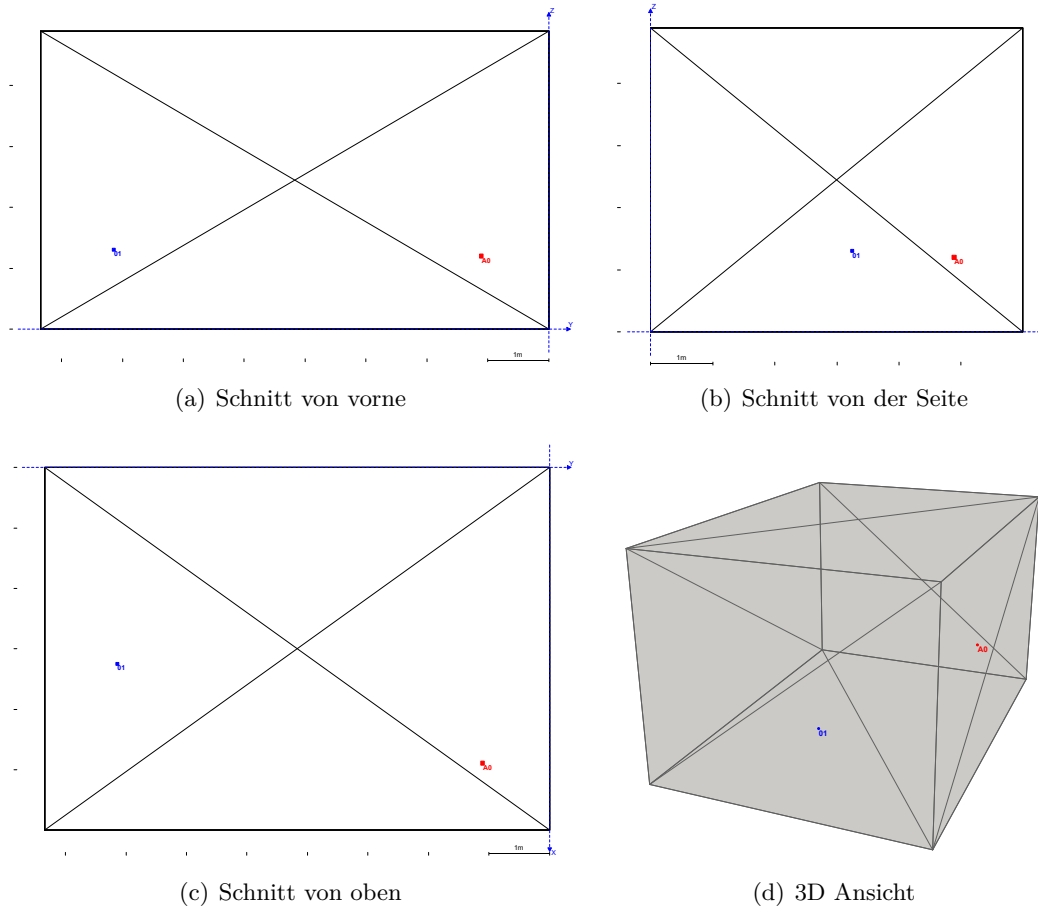


Abbildung 4.1: Hallraum der TU Graz mit den für die Simulation verwendeten Quell- (rot) und Empfängerpositionen (blau).

4.1.2 Simulationseigenschaften

	CATT	ARTWA
Quellposition	{4.89, -1.1, 1.2}	{4.89, -1.1, 1.2}
Empfängerposition	{3.25, -7.14, 1.3}	{3.25, -7.14, 1.3}
Raycount	100000	100000
Lifespan	unbekannt	1000
Luftabsorption	✓	✓
Scattering	✗	✗
Abtastrate	44100 Hz	44100 Hz
Raumtemperatur	20 °C	20 °C
Luftfeuchtigkeit	50 %	50 %

Tabelle 4.2: Simulationseigenschaften für den Hallraum.

4.1.3 Abklingkurven

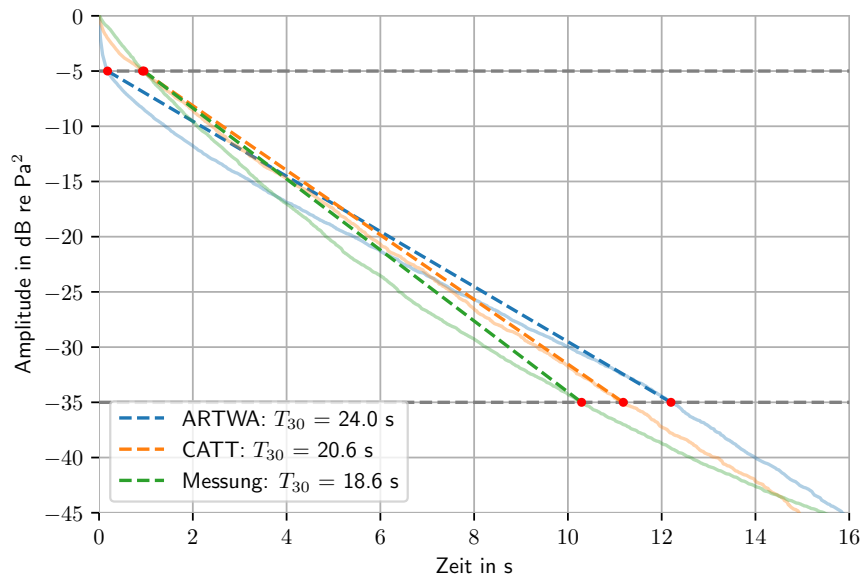


Abbildung 4.2: Aus den breitbandigen Impulsantworten erstellte Abklingkurven im Hallraum inklusive Interpolationsgeraden zur Bestimmung der T_{30} .

4.2 Absorbierender Raum

4.2.1 Raumeigenschaften

Die Raumgeometrie (Länge, Breite, Höhe) wurde bei dieser fiktiven Räumlichkeit von Kapitel 4.1 übernommen.

Materialname	Oberfläche S	α_{125}	α_{250}	α_{500}	α_{1000}	α_{2000}	α_{4000}
Mineralwolle 20 mm ohne Wandabstand	240 m ²	0.05	0.14	0.65	0.95	1.00	1.00

Tabelle 4.3: Absorptionsgrade der Wandmaterialien.

4.2.2 Simulationseigenschaften

Da dieser Raum vollständig mit Mineralwolle ausgekleidet ist und diese bei mittleren und hohen Frequenzen eine hohe Absorption aufweist, ist die Länge der erwarteten Impulsantwort jedenfalls kürzer als jene des Hallraums. Die Lifespan kann also kleiner gewählt werden, da die Lebensdauer der Strahlen nur so lange sein muss wie die erwartete Dauer der Impulsantwort.

	CATT	ARTWA
Quellposition	{4.89, -1.1, 1.2}	{4.89, -1.1, 1.2}
Empfängerposition	{3.25, -7.14, 1.3}	{3.25, -7.14, 1.3}
Raycount	100000	100000
Lifespan	unbekannt	300
Luftabsorption	✓	✓
Scattering	✗	✗
Abtastrate	44100 Hz	44100 Hz
Raumtemperatur	20 °C	20 °C
Luftfeuchtigkeit	50 %	50 %

Tabelle 4.4: Simulationseigenschaften für den absorbierenden Raum.

4.2.3 Abklingkurven

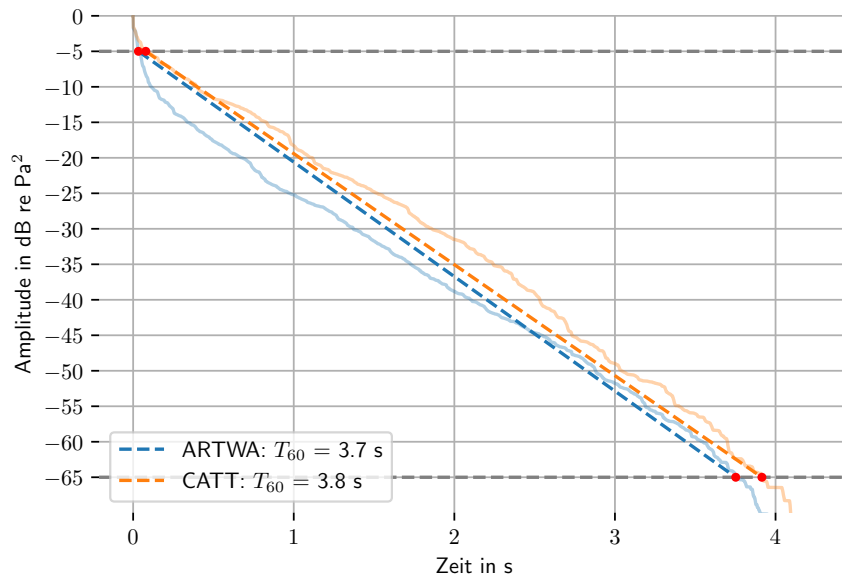


Abbildung 4.3: Aus den breitbandigen Impulsantworten erstellte Abklingkurven im absorbierenden Raum inklusive Interpolationsgeraden zur Bestimmung der T_{60} .

4.3 Konzertsaal mit gemischten Materialien

4.3.1 Raumeigenschaften

Materialname	Oberfläche S	α_{125}	α_{250}	α_{500}	α_{1000}	α_{2000}	α_{4000}
Putz 15mm verdichtet	587 m ²	0.03	0.03	0.04	0.04	0.04	0.04
Parkett auf Blindboden	324 m ²	0.20	0.15	0.10	0.10	0.05	0.10
Publikum	277 m ²	0.20	0.40	0.55	0.60	0.60	0.50
Lochplatten	150 m ²	0.38	0.73	0.49	0.47	0.37	0.33
Holzplatten Absorber	118 m ²	0.46	0.24	0.04	0.01	0.01	0.01
Beton	110 m ²	0.02	0.03	0.03	0.03	0.04	0.07
Holzpodest Podium	30 m ²	0.11	0.11	0.11	0.11	0.11	0.11

Tabelle 4.5: Absorptionsgrade der Wandmaterialien.

Die im 3D-Modell in Fig 4.4(b) rot eingezeichneten Sitzbänke entsprechen dem Material “Publikum” aus Tabelle 4.5 und weisen einen Absorptionsgrad auf, welcher sitzenden Personen auf einer gepolsterten Fläche zugeordnet werden kann.

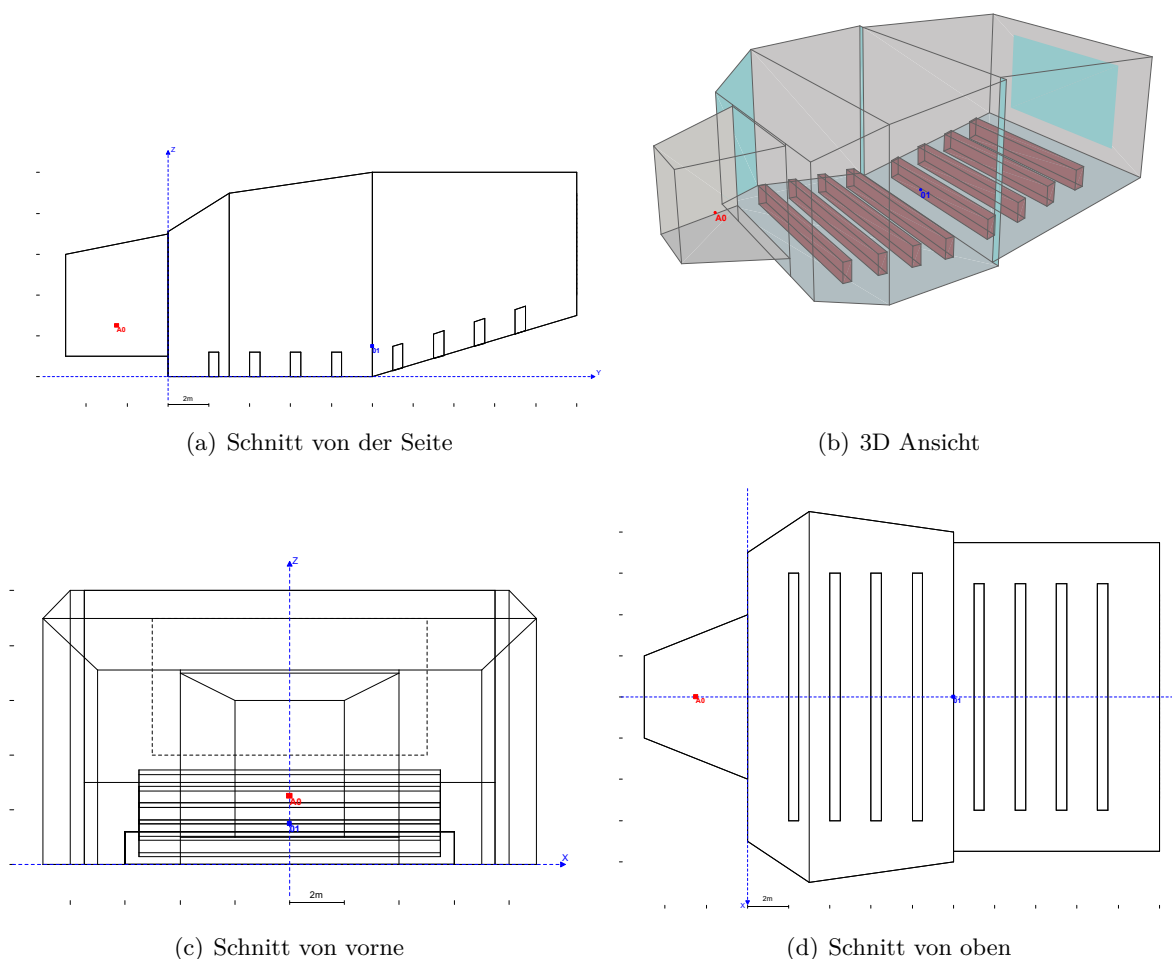


Abbildung 4.4: Fiktiver Konzertsaal mit den für die Simulation verwendeten Quell- (rot) und Empfängerpositionen (blau).

4.3.2 Simulationseigenschaften

	CATT	ARTWA
Quellposition	{0, -2.5, 2.5}	{2.5, 0, 2.5}
Empfängerposition	{0, 10, 1.5}	{-10, 0, 1.5}
Raycount	500000	500000
Lifespan	unbekannt	300
Luftabsorption	✓	✓
Scattering	✗	✗
Abtastrate	44100 Hz	44100 Hz
Raumtemperatur	20 °C	20 °C
Luftfeuchtigkeit	50 %	50 %

Tabelle 4.6: Simulationseigenschaften für den Konzertsaal.

Dieser Raum weist ein größeres Raumvolumen auf, als der Hallraum. Die Strahlen haben daher eine kleinere Wahrscheinlichkeit den Empfänger zu treffen (geringere Strahlendichte beim Empfänger). Um dies zu kompensieren, wird mit einem höheren raycount als im Hallraum simuliert.

4.3.3 Abklingkurven

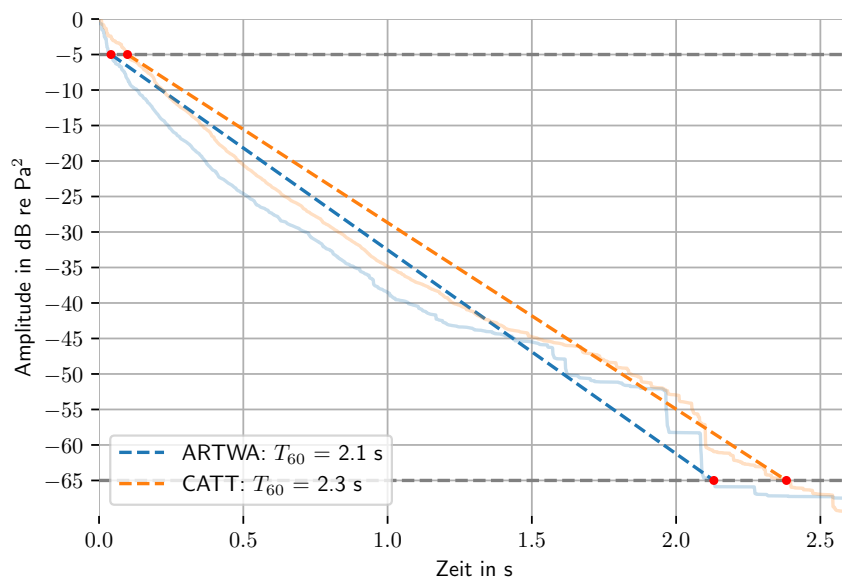


Abbildung 4.5: Aus den breitbandigen Impulsantworten erstellte Abklingkurven im Konzertsaal inklusive Interpolationsgeraden zur Bestimmung der T_{60} .

5

Diskussion

5.1 Early Decay Time

Aus den Abklingkurven in den Kapiteln 4.1.3, 4.2.3 und 4.3.3 wird ersichtlich, dass die von ARTWA generierten Impulsantworten zu einem steileren Abfall zu Beginn der Abklingkurve führen. Zum Ende hin verläuft allerdings die Abklingkurve von ARTWA wieder größtenteils parallel zu der von CATT. Bei CATT handelt es sich um ein hybrides Modell, wobei die Reflexionen niedriger Ordnungen mithilfe von Spiegelquellen berechnet werden. Der Vorteil hierbei ist, dass der Empfänger durch einen Punkt modelliert werden kann und eine Wellenfront diesen genau zu einem Zeitpunkt erreicht. Im Gegensatz hierzu muss beim Raytracing der Empfänger eine räumliche Ausdehnung haben. Genauer fungiert dabei die Querschnittsfläche der Empfängerkugel, welche orthogonal zur Einfallsrichtung des Schallstrahls steht, als Detektor. Aufgrund der durch Kugelwellenausbreitung hervorgerufenen Krümmung der Wellenfront, treffen nun die Schallstrahlen, welche eine Wellenfront repräsentieren, nicht zum exakt selben Zeitpunkt ein. Somit kommt es vor allem bei kurzen Distanzen zur Verteilung einer Wellenfront auf mehrere Abtastzeitpunkte. Diese Ungenauigkeit bei den Reflexionen niedriger Ordnung könnte eine Erklärung für den Einbruch zu Beginn der Abklingkurve sein.

5.2 Auswirkung der Soundmap

Durch die Soundmap kann vor allem bei kleinen Räumen mit kurzer Nachhallzeit eine schnelle Neuberechnung (wenige Sekunden bis zu 2 Minuten) der Empfängerposition erfolgen. Für einen Speicherzugriff auf die Festplatte wird ein vielfaches der RAM-Zugriffszeit benötigt. Somit folgt: Solange die Soundmap im RAM gespeichert werden kann, ist auch die Erstellung der Impulsantwort performant. Ein großer Nachteil der Soundmap ist daher der hohe Speicherbedarf. Aufgrund des stochastischen Raytracings ist eine große Anzahl an Strahlen erforderlich, wodurch die Anzahl der Soundmapeinträge steigt. Sobald Teile der Soundmap vom Betriebssystem auf die Festplatte ausgelagert werden müssen, verlangsamt sich die Auswertung drastisch. Die Auslagerung wird vom Betriebssystem durchgeführt, um auch anderen Prozessen RAM-Zugriff zu ermöglichen. Bei wiederholtem Zugriff auf die Soundmap ist der Verbleib im RAM wahrscheinlicher. Für eine erneute Erstellung der Impulsantwort (z.B. für eine veränderte Empfängerposition) verbessert sich die Performance folglich wieder. Wieviele Strahlen und mit welcher Lifespan man einen Raum simulieren kann, hängt somit stark von der Größe des verbauten Arbeitsspeichers ab.

5.3 Systematische Fehler

Aufgrund von sowohl physikalischen Annäherungen, als auch numerischen Fehlern kommt es bei Simulationen gezwungenermaßen zu Realitätsabweichungen. Diese Approximationen führen dazu, dass ein Modell nur in gewissen Szenarien seine Gültigkeit bewahrt. Da die geometrische Raumakustik die Ausbreitung von Schallwellen nur annähert, werden bessere Ergebnisse erzielt bei:

- kleinen Räumen
- Vermeidung von gewölbten Flächen
- hohen Absorptionsgraden
- mittleren und hohen Frequenzen

Zusätzlich sind die Kalkulationen stark von ihrem Input abhängig. Gerade die Bestimmung von Absorptionsgraden ist nicht gerade einfach und Ergebnisse stellen stets nur eine Näherung dar.

Alle vorhin genannten Parameter führen zu einer gewissen Ungenauigkeit, welche sich innerhalb einer einzigen Simulation nicht vermeiden lässt. Sollte allerdings die Simulation eines Raums einige Male wiederholt werden, wobei jedes Mal die Anzahl der Quellstrahlen (raycount) bzw. Reflexionen (lifespan) verändert und anschließend die Ergebnisimpulsantwort gemittelt wird, so können Verbesserungen erzielt werden [3].



Zusammenfassung und Ausblick

6.1 Zusammenfassung

Im Rahmen des Bachelorseminars wurde eine Software (ARTWA) entwickelt, welche es ermöglicht, 3D modellierte Räume akustisch zu simulieren. Die Simulation basiert dabei auf dem Prinzip des stochastischen Raytracings, dessen Grundlage innerhalb der geometrischen Raumakustik formuliert wird. Der Entwicklungsprozess reichte vom Erstellen einer grafischen Benutzeroberfläche inklusive 3D-Viewport, bis hin zu der Einführung einer eigens entwickelten Datenstruktur, genannt Soundmap, welche die während der Simulation erstellten Daten bei der Empfängerbewertung bereitstellt. Die Wahl der Empfängerposition ist somit komplett unabhängig vom während der Simulation stattfindenden Raytracing-Prozess und kann jederzeit variiert werden. Als Ergebnis wird die quell-/empfängerpositionsabhängige Impulsantwort in Form einer .wav-Datei exportiert. Da ein Raum über seine Impulsantwort vollständig beschrieben werden kann, kann dieser eine Vielzahl an raumakustischen Kriterien abgeleitet werden. Wie auch im Namen der Software enthalten, wird zusätzlich die Möglichkeit geboten, die akustische Situation an einer beliebigen Empfängerposition hörbar zu machen. Eine individuell geladene Sound-Datei wird somit direkt bei der Auswertung mit der jeweiligen Impulsantwort gefaltet und über den Default Audio Treiber wiedergegeben. Da die Auswertung, je nach Abtastfrequenz, in weniger als 2 Minuten (siehe Kapitel 5.2) durchgeführt wird, kann jede beliebige Position im Raum auch durch ihre Auralisation verglichen werden.

Das grafische User Interface bietet neben einigen Umweltparametern (Temperatur, Luftfeuchtigkeit) auch Einstellungen, welche die Visualisierung von Simulationsdaten erlauben. So können neben den ausgesandten Strahlen der Quelle auch jene der Reflexionen, bis hin zu einer beliebig gewählten Ordnung, dargestellt werden. Auch die Einträge der Soundmap können in Form von farbigen Punkten auf den Geometrieoberflächen sichtbar gemacht werden.

Für die Erstellung des Raummodells wurde für das Open Source 3D-Modellierungsprogramm Blender ein Addon entwickelt. Dieses kann Materialien mit deren Absorptionsgraden innerhalb einer .csv-Datei einlesen und in der Materialbibliothek von Blender speichern. Der Benutzer kann anschließend mühelos diese Materialien den Flächen seines Raums zuweisen. In Erweiterung dazu, wird vom Addon auch eine Export-Funktion zur Verfügung gestellt, welche alle für ARTWA relevanten Parameter mitberücksichtigt.

Die Software wurde auf Windows mit C++ geschrieben und wird daher als .exe-Datei bereitgestellt. Alle visuellen Aufgaben (GUI, 3D-Viewport) werden dabei direkt vom jeweils verbauten Grafikprozessor behandelt. Alle anderen Berechnungen der Simulation werden ausschließlich auf der CPU durchgeführt. Gewisse Rechenoperationen werden, soweit vom Prozessor unterstützt, zusätzlich über SIMD Instructions beschleunigt.

6.2 Ausblick

Um ARTWA zu verbessern, könnten folgende Punkte für zukünftige Arbeiten in Betracht gezogen werden.

6.2.1 Abbruchkriterium

Wie in Kapitel 1.3 erwähnt, müssen Raytracing-Simulationen immer mit einem gewissen Abbruchkriterium gekoppelt werden. Dieses wurde bei ARTWA in Form einer Variable, welche die maximale Reflexionsordnung eines Rays definiert (lifespan), berücksichtigt. Da bei einem zufällig gewählten Raum a priori nichts über diesen Wert ausgesagt werden kann und es im schlechtesten Fall zu einem frühzeitigen Abschneiden der Impulsantwort kommt, sollte die lifespan durch eine andere Abbruchbedingung ersetzt werden. Sowohl über die ungefähre Nachhallzeit (Abbruchbedingung: t_{\max}), als auch über jenen Schwellwert, ab dem die Energie eines Rays keinen entscheidenden Beitrag zum Ergebnis liefert (Abbruchbedingung: e_{\min}), können vor der Simulation Einschätzungen getroffen werden. Daher wäre es sinnvoll, eine dieser beiden Varianten zu implementieren.

6.2.2 Beugung und Scattering

Eine weiterführende Arbeit könnte das Simulieren von Beugung aufgrund eines probabilistischen Modells untersuchen. Mithilfe einer SDF (Signed Distance Function) könnten z.B. Entfernungen zu Raumbegrenzungsflächen abgeschätzt und dadurch Beugungswahrscheinlichkeiten ermittelt werden.

6.2.3 Hybride Modelle

Wie bereits in Kapitel 5 erwähnt, hat jedes Approximationsmodell für verschiedene Anwendungen Vor- und Nachteile. Sollte eine Simulation nun das Ergebnis einer realen Messung widerspiegeln, so müssen verschiedene Schallausbreitungsmodelle miteinander verknüpft werden. So kann z.B. dem Einbruch zu Beginn der Abklingkurve (Kapitel 5.1) mithilfe eines Spiegelquellenmodells für die Reflexionen niedriger Ordnung entgegengewirkt werden. Im Bereich der Welleneffekte (Beugung, Brechung) ist es zusätzlich sinnvoll, Ergebnisse von FEM / BEM Simulationen miteinzubeziehen.

6.2.4 Projektdatei

Bislang ist es zwar möglich die Ergebnisse der Simulation zu speichern, allerdings werden die vom Benutzer gesetzten Parameter dabei nicht mitgesichert. Durch die Einführung einer sogenannten Projektdatei könnte dieses Problem behoben werden. Zusätzlich könnte man die Möglichkeit schaffen, die Simulationsdaten eines Raumes als Soundmap zu exportieren. Die Empfängerposition könnte somit auch noch nach erneutem Öffnen des Programms verändert bzw. die jeweilige Impulsantwort erstellt werden, ohne die Simulation erneut starten zu müssen.