Real-Time Ultrasoundsimulation for Medical Training

Inauguraldissertation
zur Erlangung des akademischen Grades
eines Doktors der Naturwissenschaften
der Universität Mannheim

vorgelegt von

Dipl.Inf. Benny Bürger
aus Filderstadt

Mannheim, 2014

Dekan: Professor Dr. Heinz Jürgen Müller, Universität Mannheim
Referent: Professor Dr. Jürgen Hesser, Universität Heidelberg
Korreferent: Professor Dr. Wolfgang Effelsberg, Universität Mannheim

Tag der mündlichen Prüfung: 27. Mai 2015

Abstract

This thesis presents a real-time capable GPU-based ultrasound simulator suitable for medical education. Two different models are presented in the following. A 2D version has been implemented in order to simulate IVUS without modelling complex 3D geometry and a 3D version which is able to synthesize realistic looking ultrasound images in real-time. This includes ultrasound specific artifacts, which are essential for the interpretation of this data. The focus of this thesis is on the more general 3D version, that can simulate all common ultrasound modalities and is based on a convolution-enhanced ray-tracing approach which uses a deformable mesh model. This method advances the state of the art for real-time capable ultrasound simulators by following the path of the ultrasound pulse, which enables better simulation of ultrasound-specific artifacts. We evaluate our proposed method by comparison it to recent generative slicing-based strategies as well as real ultrasound images. Therefore, a gelatin ultrasound phantom containing syringes filled with different media is scanned with a real transducer. The obtained images are then compared to images which are simulated using a slicing-based technique and our proposed method. The particular benefit of our method is the accurate simulation of ultrasound-specific artifacts like range distortion, refraction and acoustic shadowing. Several benchmark scenarios are evaluated regarding simulation time, to show the performance and the bottleneck of our method. While being computationally more intensive than slicing techniques, our simulator is able to produce high-quality images in real-time, tracing over 5000 rays through mesh models with more than 2 000 000 triangles.

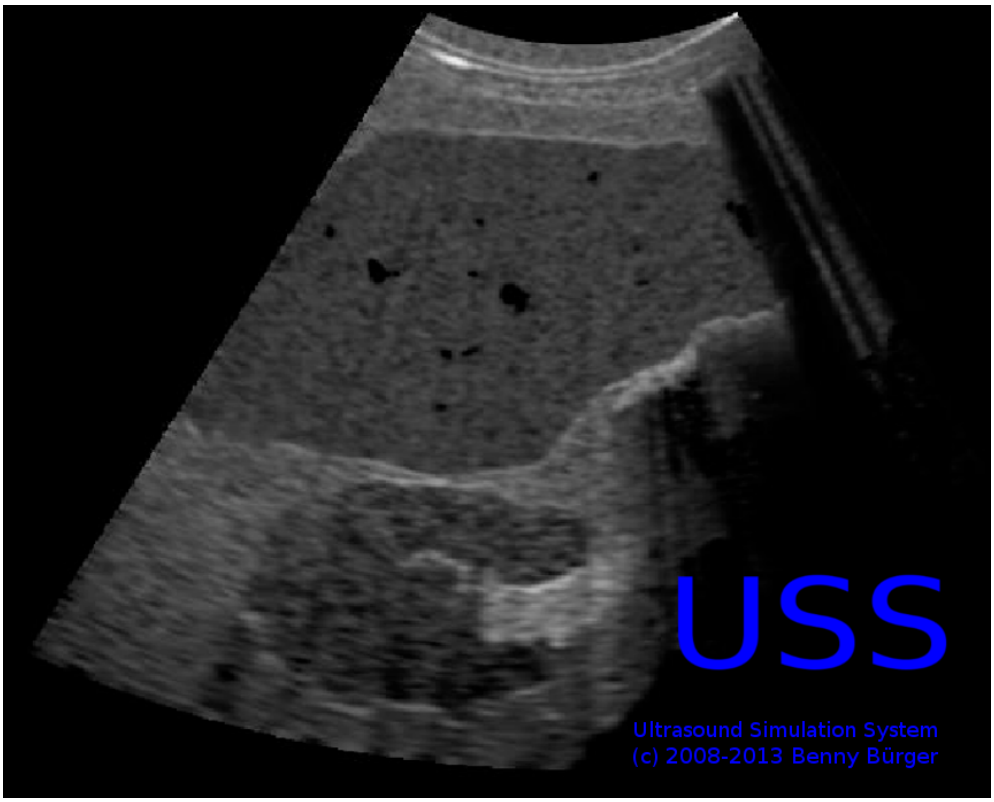
Zusammenfassung

Die vorliegende Arbeit befasst sich mit der GPU-basierten Echtzeitsimulation von Ultraschall für medizinische Ausbildung. Es werden zwei verschiedene Modelle beschrieben. Eine 2D-Version wurde implementiert, um intravaskulären Ultraschall simulieren zu können, ohne komplexe 3D-Geometrie erstellen zu müssen. Der Schwerpunkt dieser Arbeit liegt auf der 3D-Version, die künstliche, realistisch aussehende Ultraschallbilder inklusive der ultraschalltypischen Artefakte in Echtzeit simulieren kann. Die Simulation kann alle gebräuchlichen Ultraschallmodalitäten simulieren und basiert auf der Kombination einer Ray-tracing-Methode mit einem Faltungsalgorithmus. Als Geometrie werden räumlich veränderbare 3D-Gitternetze verwendet. Die vorgestellte Methode erweitert den Stand der Forschung für echtzeitfähige Ultraschallsimulatoren, indem der Weg des Ultraschallimpulses verfolgt wird, wodurch eine bessere Simulation der ultraschallspezifischen Artefakte möglich wird. Simulierte Bilder werden anhand von echten Ultraschallbildern evaluiert und mit einer aktuellen Simulationstechnik verglichen. Hierfür wurden mehrere Spritzen mit unterschiedlichen Medien gefüllt und in Gelatine eingegossen. Dieses Phantom wurde anschließend mit einem echten Ultraschallsystem durchschallt, um echte Ultraschallbilder zu erhalten. Basierend auf einem modellierten 3D-Gittermodell des Phantoms, wurden künstliche Ultraschallbilder mit der von uns beschriebenen Methode und mit einer Schnittbildmethode erzeugt. Der Vorteil der präsentierten Ray-tracing-Methode ist die realistische Simulation von Ultraschallartefakten wie Laufzeitartefakte, Spiegelartefakte oder Mehrfachechos. Mehrere Benchmarkszenarien wurden ausgewertet, um die Schnelligkeit aber auch die Grenzen der Methode aufzuzeigen. Die vorgestellte Simulationsmethode ist deutlich rechenaufwendiger als die Schnittbildmethode, trotzdem können realistische Ultraschallbilder mit mehr als 5000 Strahlen von 3D-Modellen mit über 2 000 000 Dreiecken in Echtzeit simuliert werden.

Experimentelle Strahlentherapie
Universität Heidelberg

Fakultät für Wirtschaftsinformatik
und Wirtschaftsmathematik
Universität Mannheim

Real-Time Ultrasoundsimulation for Medical Training



Danksagung

Denken und danken sind verwandte Wörter;
wir danken dem Leben, in dem wir es bedenken.

Thomas Mann (1875-1955)

In meiner Diplomarbeit habe ich mich bereits mit der Ultraschallsimulation beschäftigt und war von dem interessanten Thema gleich zu Beginn gefesselt. Während meiner anschließenden Promotionszeit in Mannheim habe ich zuerst aus Finanzierungsgründen an verschiedenen anderen Projekten gearbeitet, bis ich mich dank der Bemühungen von **Prof. Dr. rer. nat. Jürgen Hesser** schließlich ganz der Ultraschallsimulation widmen konnte. Ich habe lange Zeit gehabt, um das Themengebiet zu bedenken und möchte mich im Sinne des obigen Zitats nun bei allen Menschen bedanken, die diese Arbeit unterstützt und ermöglicht haben.

- An erster Stelle danke ich meinem Doktorvater **Prof. Dr. rer. nat. Jürgen Hesser** für die vielen hilfreichen wegweisenden Ratschläge und kritischen Reviews unserer Paper. Insbesondere auch dafür, dass er es mir ermöglichte, mich mit dem größten Teil meiner Arbeitskraft auf die Forschung zu konzentrieren.

- Meiner lieben Frau **Katarina Bürger**, ohne deren wachsames Auge meine Veröffentlichungen sicher einige orthographische Fehler mehr enthalten hätten. Sie wies auf Schwächen hin und konnte als Fachfremde immer wieder zeigen, wo noch Erklärungsbedarf bestand.

- Meinen Eltern **Marlis Bürger-Ulbricht** und **Erich Bürger**, für die moralische Unterstützung und insbesondere meiner Mutter für die unzähligen Stunden, die sie Korrektur gelesen hat.

- Meinen Freunden und Kollegen an der Universität und der CATHI GmbH, die mir bei Fachfragen stets hilfreich zur Seite standen.

- Meinen Kollegen am Universitätsklinikum, die mich in die Bedienung von Ultraschallgeräten einwiesen und mir ihr Fachwissen, Ultraschallbilder und CT-Volumen zur Verfügung stellten.

Akronyme und Fachwörter[1]

Black Box Ein geschlossenes System unter Vernachlässigung des inneren Aufbaus.

BVH Hüllkörper (engl. **B**ounding **V**olume **H**ierarchy)

Code Kurzform von Programmcode

Computercluster engl. für Rechnerverbund

CR klassisches Röntgen: ein auf Röntgenstrahlen basierendes bildgebendes Verfahren, mit dem das zu untersuchende Objekt durchleuchtet werden kann.

CT Computertomographie: ein auf Röntgenstrahlen basierendes bildgebendes Verfahren, mit dem das zu untersuchende Objekt in Querschnittbildern dargestellt werden kann.

Face engl. für Fläche. Im Kontext dieser Arbeit wird damit ein Polygon bezeichnet, das durch mehrere Eckpunkte miteinander verbunden ist.

GPGPU Allzweck-Berechnung auf Grafikprozessoreinheiten (eng. **G**eneral **P**urpose Computation on **G**raphics **P**rocessing **U**nit)

GPU Grafikprozessor (engl. **G**raphics **P**rocessing **U**nit)

IVUS Intravaskulärer Ultraschall ist ein Ultraschallverfahren, bei dem der Schallkopf durch Gefäße geführt wird, um diese zu beurteilen.

Material siehe Medium

Medium Das Medium oder Ausbreitungsmedium bezeichnet in der Wellenlehre die Substanz, welche von den Wellen durchquert wird.

Mesh-Modell Fachbezeichnung eines Oberflächenmodells auf der Basis eines Polygonnetzes.

Modalität Oberbegriff für medizinische Geräte, die für bildgebende Verfahren in der medizinischen Diagnostik eingesetzt werden.

MR Siehe MRT.

[1] Hier aufgelistete computerspezifische Fachwörter werden im Folgenden in der englischen Variante verwendet.

MRT Magnetresonanztomographie: ein auf Magnetfeldern basierendes bildgebendes Verfahren, mit dem das zu untersuchende Objekt in Querschnittbildern dargestellt werden kann.

PET **P**ositronen-**E**missions-**T**omographie: ein auf radioaktiven Substanzen basierendes bildgebendes Verfahren, mit dem das zu untersuchende Objekt in Querschnittbildern dargestellt werden kann.

Programmcode gängiges Fachwort für Quelltext

Ray-casting engl. für vereinfachte Strahlverfolgung, Reflexionen und Brechungen werden nicht beachtet

Ray-tracing engl. für komplexe Strahlverfolgung unter Einbeziehung von Reflexionen und Brechungen. Jeder Strahl kann in mehrere Teilstrahlen aufgespalten werden.

Speckles helle und dunkle Fleckenmuster im Ultraschall, entstanden durch Interferenz von Schallwellen. Speckle bezeichnet sowohl das komplette Interferenzmuster, als auch einen einzigen Fleck.

Szenarium Im Kontext dieser Arbeit wird hiermit ein komplettes Modell (Geometrie und Medieneinstellungen) bezeichnet, mit dem Ultraschallbilder simuliert werden können.

TEE Transösophageale Echokardiographie, ist ein Ultraschallverfahren bei dem der Schallkopf durch die Speiseröhre durchgeführt wird, um Untersuchungen am Herz durchzuführen.

US Diagnostischer Ultraschall: ein auf Laufzeitmessung von Ultraschall basierendes bildgebendes Verfahren, mit dem das zu untersuchende Objekt in Schnittbildern dargestellt werden kann.

Vertex engl. für Eckpunkt. Im Kontext dieser Arbeit ist ein Vertex immer dreidimensional $(x, y, z) \in \Re$ und beschreibt i.d.R. den Eckpunkt einer Fläche.

Inhaltsverzeichnis

1 Einleitung **1**
- 1.1 Ziele der Arbeit . . . 3
  - 1.1.1 Speckle-Rauschen . . . 3
  - 1.1.2 Gewebegrenzen . . . 4
  - 1.1.3 Ultraschallartefakte . . . 4
  - 1.1.4 Echtzeitfähigkeit . . . 4
  - 1.1.5 Modell der Geometrie . . . 4
  - 1.1.6 Ultraschallsysteme . . . 4
  - 1.1.7 Bilddarstellung . . . 5
  - 1.1.8 Schallkopfsteuerung . . . 5

2 Grundlagen **7**
- 2.1 Geschichte des diagnostischen Ultraschalls . . . 7
- 2.2 Physik des Ultraschalls . . . 8
  - 2.2.1 Schall . . . 8
  - 2.2.2 Schallfeldgrößen . . . 9
  - 2.2.3 Reflexion und Brechung . . . 11
  - 2.2.4 Absorption . . . 12
  - 2.2.5 Streuung . . . 12
  - 2.2.6 Piezoelektrischer Effekt . . . 14
- 2.3 Bildgebung . . . 14
  - 2.3.1 Echo-Impuls-Verfahren . . . 14
  - 2.3.2 Aufnahmezeit . . . 15
  - 2.3.3 Schallfeld . . . 16
  - 2.3.4 Darstellungsmethoden . . . 18
  - 2.3.5 Ultraschallköpfe . . . 21
  - 2.3.6 Artefakte . . . 23

3 Stand der Forschung **29**
- 3.1 Simulationssysteme . . . 29
  - 3.1.1 IVUS . . . 29
  - 3.1.2 Brachytherapie . . . 29
  - 3.1.3 Sonographie . . . 30
  - 3.1.4 Ultraschallsimulationen . . . 31
  - 3.1.5 Field II . . . 31

3.1.6 k-Wave . . . 32
3.1.7 Ultrasim . . . 32
3.1.8 Wave2000, Wave2500 und Wave3000 . . . 32
3.2 Verfahren zur Simulation von Ultraschall . . . 32
3.2.1 Das Schnittbildverfahren . . . 32
3.2.2 Generative Verfahren . . . 33

4 Realisierung **39**
4.1 2D-Ultraschall Simulation . . . 39
4.1.1 Modelle . . . 39
4.1.2 Simulation . . . 40
4.2 3D-Ultraschall Simulation . . . 41
4.2.1 Modelle . . . 41
4.2.2 Physik des Ray-tracings . . . 42
4.3 Nachbearbeitung . . . 49

5 Implementierung **53**
5.1 Programmdesign . . . 54
5.1.1 Eigene Software-Bibliotheken . . . 54
5.2 CUDA C . . . 56
5.3 OptiX . . . 56
5.3.1 Beschleunigungsstrukturen . . . 57
5.3.2 Schnittpunktalgorithmen . . . 59
5.3.3 Implementierungsdetails fürs Ray-tracing . . . 59
5.4 Erweiterungen für eine Brachytherapiesimulation . . . 61
5.4.1 Deformation des inneren Gewebes . . . 61
5.4.2 Materialtypen . . . 63
5.5 Koordinatentransformation . . . 63
5.6 Performanzoptimierung . . . 65
5.6.1 Threads und Synchronisation . . . 65
5.6.2 Funktions- und Stabilitätsoptimierungen . . . 67

6 Modellerzeugung **69**
6.1 2D-Modelle . . . 69
6.2 3D-Modelle . . . 70
6.2.1 Nachbearbeitung von Mesh-Modellen . . . 71
6.3 Parameterschätzung . . . 73
6.4 Szenarienbeschreibung . . . 77
6.4.1 Medium- und Objekteigenschaften der Szene . . . 80
6.4.2 Schallkopf-, Simulation- und sonstige Parameter . . . 81

7 Resultate **83**
7.1 2D 83
7.1.1 Realitätsnähe 83
7.1.2 Performanz 84
7.2 3D 85
7.2.1 Resultate 85

8 Zusammenfassung und Ausblick **99**
8.1 Ausblick 102

Literaturverzeichnis **103**

1 Einleitung

Diagnostischer Ultraschall (US) ist seit Mitte der 1980er Jahre nicht mehr aus der modernen Medizin wegzudenken. Neben der Röntgendiagnostik (CR) ist es das wichtigste bildgebende Verfahren der Medizin. Diagnostischer Ultraschall bietet sehr viele Vorteile gegenüber den anderen Verfahren wie Computertomographie (CT), Positronenemissionstomographie (PET) oder Magnetresonanztomographie (MRT):

- kostengünstig - US kostet nur einen Bruchteil von anderen Modalitäten.
- echtzeitfähig - US kann mehrere Bilder pro Sekunden aufnehmen, im Gegensatz zu CT- oder MRT-Verfahren.
- unschädlich - US hat im Gegensatz zu CR und CT keine Strahlenbelastung und kann dadurch seine Echtzeitfähigkeit voll nutzen.
- portabel - heutige US-Systeme sind kleiner als ein Aktenkoffer, wohingegen die übrigen Modalitäten i.d.R. fest in Räume integriert sind.
- hohe Auflösung - US kann eine deutlich höhere Auflösung erreichen als die übrigen Modalitäten.

Aufgrund all dieser Vorteile ist es für Laien zunächst verwunderlich, dass auch weiterhin teure und gesundheitsschädliche Modalitäten eingesetzt werden. Dies liegt zum einen daran, dass sich die einzelnen Modalitäten ergänzen und nicht ersetzen. Das heißt, in bestimmten Modalitäten sind bestimmte Befunde besser zu erkennen als in anderen. Des Weiteren besitzt auch die Modalität US einige Nachteile:

- frequenzabhängige Auflösung - die Auflösung von US-Bildern ist abhängig von der Frequenz des Ultraschallpulses; hohe Frequenz $\rightarrow$ hohe Auflösung.
- eingeschränkte Reichweite - die Reichweite des US ist ebenso abhängig von der Frequenz des Ultraschallpulses; hohe Frequenz $\rightarrow$ niedrige Reichweite.
- schwierige Interpretation - US zeigt nicht die exakte Geometrie der Organe, sondern den Verlauf der Schallwellen und enthält viele Artefakte.

Die Auflösung und Reichweite von Ultraschallbildern kann zwar durch bessere Geräte und neue Ansteuerungsmethoden optimiert werden, jedoch nicht über die Grenzen des physikalisch Möglichen hinaus. Auch die Qualität der Bilder kann zwar durch bessere Geräte und Bildbearbeitungsmethoden erhöht werden, die typischen Artefakte und Eigenschaften von US-Bildern werden jedoch auch künftigen Medizinern erhalten bleiben. Zwar gibt es bereits seit mehreren Jahren dreidimensionalen Ultraschall (3D-US), aber dieser wird aus Kostengründen hauptsächlich in der Pränataldiagnostik verwendet.

Damit US-Bilder von Medizinern richtig interpretiert werden können, muss die Qualität der diagnostischen Ausbildung ein hohes Niveau haben. Die heutige Ausbildung basiert auf Bildern, Videos und echten Ultraschalluntersuchungen. Immer häufiger kommen auch Ultraschallsimulatoren zum Einsatz. Bilder und Videos besitzen den großen Nachteil, dass der Mediziner nicht selbst den Schallkopf halten kann, wodurch der Lerneffekt gering ist. Die Möglichkeit echte Ultraschalluntersuchungen an Freiwilligen durchzuführen ist ideal, jedoch ist der Mediziner hierbei immer auf freiwillige Versuchspersonen angewiesen und sieht in der Regel nur Standardfälle ohne besonderen Befund. Auch Operationen wie ultraschallgeführte Nadeleinführungen können nicht an echten Patienten trainiert werden. Daher werden immer öfter Ultraschallsimulatoren verwendet, die eine große Anzahl von Untersuchungen mit unterschiedlichen Befunden simulieren können. Die zur Zeit in der Praxis genutzten Simulatoren lassen sich in zwei Kategorien einteilen.

Kategorie A benutzt ein echtes Ultraschallsystem und einen Ultraschalldummy, im Folgenden auch Ultraschallphantom oder kurz Phantom genannt. Dieser Dummy ist, in Bezug zu den Schalleigenschaften, mit gewebeähnlichen Materialien gefüllt. Die Dummys sind teuer und für jedes neues Fallbeispiel muss ein neues Phantom angeschafft werden. Dieses Verfahren eignet sich nicht zur Simulation von ultraschallgeführter Nadeleinführung, da die empfindlichen Phantome im Laufe der Zeit durch die Nadel beschädigt würden und häufig ausgetauscht werden müssten.

Kategorie B benutzt eine Schallkopfattrappe, welche die aktuelle Orientierung des Schallkopfes an ein Simulationssystem weitergibt. Anhand der Orientierung wird dann auf der Basis eines echten 3D-Ultraschallvolumens ein 2D-Ultraschallbild generiert. Dieser Typus von Simulator besitzt den Nachteil, dass viele der typischen Ultraschallartefakte nicht simuliert werden. In Kapitel 3 wird dieser Typ von Ultraschallsimulatoren genauer beschrieben.

Im Laufe dieser Arbeit wurde ein neues Verfahren entwickelt, um in Echtzeit künstliche Ultraschallbilder zu generieren. Es ermöglicht die Simulation von realistischen Ultraschallbildern, insbesondere der typischen Ultraschallartefakte. Das Verfahren kann angewendet werden, um beliebige Arten von Ultraschallbildern zu simulieren. Neben Standarduntersuchungen, wie Untersuchungen des Bauchraumes (Abdomen-Sonographie) mit Hilfe von Linear- und Konvex Schallköpfen, können auch weniger gebräuchliche Sonographiearten wie intravaskulärer

Ultraschall (IVUS) und transösophageale Echokardiographie (TEE) simuliert werden.

Die nachfolgende Arbeit ist folgendermaßen aufgebaut:

Zunächst werden in diesem Kapitel die Ziele der Arbeit definiert. Im folgenden Kapitel werden die Grundlagen für die restliche Arbeit erklärt. Anschließend wird der aktuelle Stand der Forschung zur Ultraschallsimulation beschrieben, wobei der Schwerpunkt auf echtzeitfähigen Simulationen liegt. In Kapitel 4 wird die Funktionsweise der selbst entwickelten Methode zur Echtzeit Ultraschallsimulation beschrieben. Kapitel 5 beschreibt Implementierungsdetails des Ultraschallsimulators. Verfahren mit denen Modelle für die Simulation erzeugt werden können, werden in Kapitel 6 erläutert. Die Qualität der Bilder und die Geschwindigkeit der Simulation werden in Kapitel 7 untersucht. Eine abschließende Zusammenfassung und ein Ausblick erfolgen in Kapitel 8.

1.1 Ziele der Arbeit

Der Schwerpunkt der vorliegenden Arbeit liegt in der Entwicklung einer echtzeitfähigen Ultraschallsimulation. Das simulierte Ultraschallbild soll soweit wie möglich mit einem echten Ultraschallbild übereinstimmen. Die folgenden Eigenschaften eines Ultraschallbildes sind hierbei relevant: Speckle-Rauschen, Gewebegrenzen und Ultraschallartefakte. Die Bilder sollen in Echtzeit erzeugt werden, unabhängig von der Lage des simulierten Schallkopfes. Das genutzte Modell zur Beschreibung der Geometrie des simulierten Objektes muss deformierbar sein, ohne dass die Echtzeitfähigkeit beeinträchtigt wird. Die Simulation soll in der Lage sein verschiedene Ultraschallsysteme zu simulieren.

Existierende Verfahren sind in der Regel zu rechenaufwendig oder zu stark reduziert, als dass sie für die Simulation geeignet wären. Daher wurde ein eigenes Verfahren entwickelt, um die gesteckten Ziele zu erreichen. Im Folgenden werden diese Ziele genauer erklärt und weitere wichtige Punkte für einen medizinischen Ultraschallsimulator aufgezeigt.

1.1.1 Speckle-Rauschen

Das Speckle-Rauschen entsteht durch Interferenz der Schallwellen und zeichnet sich durch die auf kurze Distanz abwechselnden hellen und dunklen Flecken (Speckles) im Ultraschallbild ab. Die simulierten Speckles sollen den Echten möglichst ähnlich in Bezug auf Größe, Verteilung und Intensität sein. Des Weiteren sollen die Speckles zwar zufällig verteilt sein, aber sie sollen sich bei einer Bewegung des Schallkopfes realistisch verhalten. Wird der Schallkopf parallel versetzt, soll sich das Aussehen des Speckle-Rauschens nur minimal ändern. Bei einer parallelen Verschiebung des Schallkopfes sollen die Speckles in die Richtung der Verschiebung wandern.

1.1.2 Gewebegrenzen

Die Gewebegrenzen der simulierten Bilder sollen möglichst identisch zu den echten Ultraschallbildern sein. Die wichtigsten Punkte sind hierbei die Lage der Gewebegrenzen auf dem Ultraschallbild und die Intensität. Diese ist nicht nur von den Geräten und Gewebeeigenschaften abhängig, sondern auch von dem Winkel des Schalleinfalles.

1.1.3 Ultraschallartefakte

Der Simulator soll alle wichtigen Ultraschallartefakte simulieren: das Schichtdickenartefakt, die Schallabschattung, die Schallverstärkung, die Spiegelung, das Mehrfachecho- und verschiedene Laufzeitartefakte. Da in diesem Punkt der größte Schwachpunkt aktueller Ultraschallsimulatoren liegt, wird diesem Ziel besondere Aufmerksamkeit gewidmet.

1.1.4 Echtzeitfähigkeit

Die Simulation soll in der Lage sein Ultraschallbilder in Echtzeit zu erzeugen. Über die genaue Definition von Echtzeitfähigkeit lässt sich sicherlich streiten. Teilweise wird darunter eine Bildwiederholungsrate von 15 Bildern pro Sekunde verstanden, da eine höhere Bildfrequenz visuell nicht erkennbar ist. Andere Definitionen gehen von einer Bildwiederholungsrate von 25 Bildern pro Sekunde oder der Rate, mit der Ultraschallgeräte arbeiten,[1] aus. Wichtig für einen medizinischen Simulator ist eine ruckelfreie Darstellung des Bildes, was bei ungefähr 15 Bildern pro Sekunde der Fall ist.

1.1.5 Modell der Geometrie

Das Modell, das zur Beschreibung der Geometrie des zu schallenden Bereichs verwendet wird, soll in der Lage sein, beliebige, in der Natur vorkommende Geometrien darzustellen. Außerdem soll es möglichst leicht deformierbar und modifizierbar sein. Daher muss es sich um ein gängiges Modell zur Beschreibung von Geometrie handeln. Alternativ können auch exotische Modelle verwendet werden, falls Algorithmen existieren, die das Modell in ein gängiges Modell konvertieren können.

1.1.6 Ultraschallsysteme

Es sollen alle möglichen Arten von Ultraschallsystemen simulierbar sein. Mögliche Bespiele sind intravaskulärer, transösophagealer, transrektaler oder abdominaler Ultraschall. Der Nutzer muss nur die Eigenschaften des Schallkopfes verändern

[1] Ultraschallgeräte können eine Bildrate von über 100 Bildern pro Sekunde erreichen.

und die Geometrie des zu schallenden Bereiches definieren, um den Ultraschall seiner Wahl zu simulieren. Die Steuerung des Ultraschallkopfes muss natürlich fallspezifisch angepasst werden.

1.1.7 Bilddarstellung

Das künstliche Ultraschallbild soll auf einem Monitor angezeigt werden. Die üblichen Nachbearbeitungsalgorithmen von echten Ultraschallsystemen sollen auch implementiert werden. Parameter, die vom Anwender in der Realität beeinflusst werden können, sollen auch im Simulator vom Anwender in Echtzeit veränderbar sein.

1.1.8 Schallkopfsteuerung

Der Anwender soll den Schallkopf wie in der Realität bewegen. Beim abdominalen Ultraschall muss der Schallkopf frei dreh- und positionierbar sein. Beim intravaskulärem, transösophagealem und transrektalem Ultraschall wird der Schallkopf über eine Schiebe- und Drehbewegung gesteuert. Optional soll dem Anwender eine Rückstellkraft vom Schallkopf mitgeteilt werden.

2 Grundlagen

Zum besseren Verständnis der nachfolgenden Arbeit werden hier die Grundlagen der medizinischen Ultraschallbildergenerierung beschrieben. Nach einem kurzen Überblick über die Geschichte des diagnostischen Ultraschalls, werden die zur Ultraschallbildgebung relevanten physikalischen Grundlagen erläutert. Anschließend wird näher auf die Technik der Bildgenerierung eingegangen. Im Anschluss daran werden die verschiedenen Ultraschallartefakte beschrieben.

2.1 Geschichte des diagnostischen Ultraschalls

Die mathematischen Grundlagen für medizinischen Ultraschall wurden im späten 19. Jahrhundert durch den Engländer Lord Rayleigh mit seiner Arbeit *The Theory of Sound* gelegt. Basierend auf den dort veröffentlichten Prinzipien und dem von den Gebrüdern Curie entdecktem Piezo-Effekt wurde am Anfang des letzten Jahrhunderts das Sonar entwickelt. Das Sonar ist ein Verfahren zur Ortung von Gegenständen im Raum und unter Wasser mittels ausgesandter Schallimpulse. Dass dieses Verfahren bereits seit Jahrtausenden von der Fledermaus zur Orientierung genutzt wird, wurde erst in den späten 1930er Jahren durch Donald Griffin entdeckt und in den frühen 1940er Jahren durch Sven Dijkgraaf experimentell nachgewiesen.

Ende der 1920er Jahre entwickelte Sergei Y. Sokolov eine Durchschallungsmaschine, um Materialfehler zu entdecken. Allerdings war die Technik noch nicht ausgereift genug, um sinnvoll eingesetzt werden zu können. Sokolov entwickelte ebenfalls ein Mikroskop auf Basis von Ultraschallwellen, das jedoch erst gegen Ende der 1930er Jahre umgesetzt werden konnte. Vorher war die nötige Technologie zur Erzeugung der Ultraschallimpulse noch nicht verfügbar.

Im gleichen Zeitraum entwickelte Karl T. Dussik eine Methode, um Ultraschall für medizinische Diagnostik nutzen zu können. Jedoch zeigte sich bereits hier, dass man für die Anwendung von diagnostischem Ultraschall die physikalischen Grundlagen kennen und berücksichtigen muss. Dussik wollte mit seiner Methode das menschliche Gehirn untersuchen, das aufgrund der Schädelknochen wohl am schwierigsten zu durchschallende menschliche Organ.

In den 1950er Jahren wurde von Howry und Bliss ein Wasserbadscanner entwickelt. Kurz darauf wurde 1954 der *„compound-scanner“* entwickelt, der in der Lage war, zweidimensionale Bilder zu erzeugen. Da der Patient jedoch komplett im Wasser sein und teilweise mit Bleiplatten beschwert werden musste, war das Gerät

nur sehr bedingt für kranke Menschen einsetzbar. Drei Jahre später wurde dieses Prinzip von I. Donald weiterentwickelt. Mit seinem *„contact-compound-scanner“* konnte der Schallkopf direkt auf die Haut aufgesetzt werden.

Mitte der 1960er Jahre entwickelte Siemens das Vidoson, ein Ultraschallgerät, das in der Lage war, 2D-Echtzeitaufnahmen zu liefern. Die Australier Kossoff und Garret entwickelten Anfang der 1970er Jahre die *„grey-scale-technique“*. Diese Technik, stellt die Sonogramme als Graustufenbilder dar und löste die bis dahin genutzte Darstellung der Bilder in schwarz und weiß ab. Als kurz darauf Ende der 1970er Jahre der erste Sektorscanner serienreif war, kam der große Durchbruch der Sonographie. Weiterführende Informationen über die Geschichte des diagnostischen Ultraschall findet man im Ultraschallmuseum der Deutschen Gesellschaft für Ultraschall in der Medizin (DEGUM) (DEG).

2.2 Physik des Ultraschalls

In den folgenden Abschnitten werden nur die für das Verständnis dieser Arbeit notwendigen physikalischen Grundlagen des Ultraschalls erläutert. Für eine detailliertere physikalische Beschreibung des Ultraschalls und seinen Anwendungen sei auf das Buch *Physik und Technik des Ultraschalles* von H. Kuttroff verwiesen (Kut88).

Ultraschallwellen sind Schallwellen mit Frequenzen außerhalb des menschlichen Hörspektrums. Dieses ist individuell verschieden und abhängig vom Lebensalter. Im Allgemeinen können Frequenzen von mehr als 20 kHz nicht mehr mit dem menschlichen Gehör wahrgenommen werden. Der Frequenzbereich ab 1 GHz wird in der Fachliteratur als Hyperschall bezeichnet. Ultraschall besitzt eine Vielzahl von Anwendungsgebieten und kann sowohl zur Untersuchung, als auch zur Manipulation von Objekten verwendet werden. Zu den bedeutendsten Anwendungsgebieten zählen die zerstörungsfreie Werkstoffuntersuchung (engl. *NDE* für ***n**on**d**estructive **e**valution of materials*) und die Ultraschalldiagnostik der Medizin (Sonographie und Echographie). Für den diagnostischen Ultraschall werden Frequenzen von 1 MHz bis 40 MHz verwendet.

Physikalisch gesehen unterscheiden sich Ultraschallwellen nicht von normalen Schallwellen, für den Rest der Arbeit wird der Begriff Schall und Ultraschall daher synonym verwendet. Aus geometrischer Sicht kann die Schallwelle als Lichtstrahl behandelt werden. Die Gesetze zur Berechnung der Schallausbreitung sind dann identisch mit den Gesetzmäßigkeiten der geometrischen Optik.

2.2.1 Schall

Schall ist eine mechanische Schwingung in einem Schallübertragungsmedium, im Folgenden kurz Medium oder Material m genannt, die sich auf Grund der elastischen Kopplung in diesem ausbreitet. In der Physik wird diese Schwingung

als Welle bezeichnet, die sich in Longitudinal- und Transversalwellen einteilen lässt. Erstere schwingen in Richtung der Ausbreitungsrichtung und letztere senkrecht zur Ausbreitungsrichtung. Während sich Longitudinalwellen in festen, flüssigen und gasförmigen Medien in alle Raumrichtungen ausbreiten können, werden sie häufig als Kugelwellen bezeichnet. Dagegen breiten sich Transversalwellen nur in Festkörpern aus und sind in der Ultraschalldiagnostik nicht von Bedeutung. Die Ausbreitung der Kugelwellen wird durch die akustischen Gesetze beschrieben. Diese sind eine Approximation erster Ordnung. Nichtlinearitäten werden von ihnen nicht berücksichtigt.

Ultraschallwellen unterliegen während ihrer Ausbreitung in biologischem Gewebe unterschiedlichen Wechselwirkungen mit dem Medium. Die auftretenden Effekte sind hierbei Reflexion, Beugung, Absorption und Streuung, welche in den Abschnitten 2.2.3 bis 2.2.5 näher erläutert werden. Aufgrund dieser charakteristischen Eigenschaften entsteht später das Ultraschallbild. Verschiedene homogene Gewebetypen lassen sich so mit Kenntnis der Ultraschallphysik voneinander unterscheiden. Homogen bezieht sich hierbei auf die makroskopische Ebene[1], denn mikroskopisch[2] gesehen sind auch diese gleichen Gewebetypen inhomogen.

2.2.2 Schallfeldgrößen

Die Schallwellen breiten sich im durchschallten Medium mit einer bestimmten Geschwindigkeit, im folgenden Schallgeschwindigkeit c $[\frac{\mathrm{m}}{\mathrm{s}}]$ genannt, aus. Die Schallgeschwindigkeit ist von der Wellenlänge λ, der Frequenz f, der Temperatur und dem Medium abhängig. Es gilt die Beziehung $c = \lambda f$. Weiterhin gelten in Abhängigkeit der Art des Mediums folgende Beziehungen:

Medien	Formel	Konstanten der Medien
Gase	$c = \sqrt{\frac{\kappa \cdot p_{stat}}{\rho}}$	κ Isentropenexponent, p_{stat} Gasdruck
Festkörper	$c = \sqrt{\frac{E}{\rho}}$	E Elastizitätmodul
Flüssigkeiten	$c = \sqrt{\frac{1}{\kappa \cdot \rho}}$	κ Kompressibilität

Die spezifische Dichte ρ $[\frac{\mathrm{kg}}{\mathrm{m}^3}]$ gibt an, wie viel Masse sich in einem homogenen Medium befindet. Der Isentropenexponent κ beschreibt das Verhältnis zwischen der spezifischen Wärme bei konstantem Druck zur spezifischen Wärme bei konstantem Volumen. Das Elastizitätsmodul E $[\frac{\mathrm{kg}}{\mathrm{m} \cdot \mathrm{s}^2}]$ beschreibt den Zusammenhang zwischen Spannung und Dehnung von festen Medien. Analog hierzu beschreibt die Kompressibilität κ $[\frac{\mathrm{m} \cdot \mathrm{s}^2}{\mathrm{kg}}]$ die Komprimierbarkeit von flüssigen Medien.

Der Schalldruck p bezeichnet die Druckschwankungen, die bei der Ausdehnung von Schall innerhalb eines Mediums auftreten. Er errechnet sich aus der Differenz zwischen dem Gesamtdruck und dem statischen Druck $p = p_{ges} - p_0$ und wird in

[1] makroskopischen Betrachtungsweise: die Eigenschaften eines Systems anhand sinnvoller statistischer Größen herleiten

[2] mikroskopische Betrachtungsweise: die Eigenschaften eines Systems anhand seiner kleinsten Bestandteile herleiten

Pascal [$\frac{\mathrm{N}}{\mathrm{m}^2}$] angegeben. Die Ausbreitung des Schalldruckes wird durch die lineare Wellengleichung beschrieben.

$$\left(\frac{\partial}{\partial x}\right)^2 p(x,t) = \frac{1}{c^2}\left(\frac{\partial}{\partial t}\right)^2 p(x,t) \tag{2.1}$$

Der Term $\left(\frac{\partial}{\partial x}\right)^2$ beschreibt hierbei die zweite Ableitung nach dem Ort und $\left(\frac{\partial}{\partial t}\right)^2$ die zweite Ableitung nach der Zeit.

Eine allgemeine Lösung der Wellengleichung ist gegeben durch $p(x,t) = p_0 \cdot g(t \pm \frac{x}{c})$, wobei x der Abstand in der räumlichen Ausbreitungsrichtung der Schallwelle ist. Ersetzt man die allgemeine Funktion g durch eine sinusförmige Elementarwelle mit Frequenz f und Phase ϕ so erhält man:

$$p(x,t) = p_0 \sin\left(2\pi f\left(t - \frac{x}{c}\right) + \phi\right) \tag{2.2}$$

Da sich jedes Schallfeld als Überlagerung von solchen sinusförmigen Elementarwellen darstellen lässt, kann eine Welle mit Hilfe von Gleichung 2.2 vollständig beschrieben werden. Für weiterführende Informationen zur Wellengleichung sei auf *An Introduction to Acoustics* verwiesen (RH04).

Die Schallkennimpedanz (auch Schallimpedanz, Schallwiderstand oder Schallwellenwiderstand genannt) ist definiert als das Produkt der Dichte und der Schallgeschwindigkeit.

$$Z = \rho c \tag{2.3}$$

Wie in den nächsten Abschnitten zu sehen ist, beeinflusst die Schallkennimpedanz maßgeblich die Absorption, Brechung und Reflexion.

Zur Beschreibung der Energie eines Schallfeldes wird die Schallfeldgröße Schallenergie W [J] verwendet. Meistens ist nicht die gesamte Energie von Interesse, sondern nur die Energie an einem bestimmten Punkt. Diese Schallenergiedichte E ist als Schallenergie an einem bestimmten Ort $\overrightarrow{x}$ definiert als:

$$E = \frac{\partial W(\overrightarrow{x})}{\partial V} \tag{2.4}$$

V bezeichnet das Volumen, wodurch sich als Einheit für E [$\frac{\mathrm{J}}{\mathrm{m}^3}$] ergibt.

In Tabelle 2.1 sind typische Schallfeldgrößen des menschlichen Körpers zu sehen. Da im menschlichen Körper eine annähernd konstante Temperatur herrscht, wird die Abhängigkeit von der Temperatur vernachlässigt und die Temperatur für jedes Medium als gleich angenommen.

Substanz	$c[\frac{\mathrm{m}}{\mathrm{s}}]$	$\rho[\frac{\mathrm{g}}{\mathrm{cm}^3}]$	$Z[\frac{\mathrm{g}}{\mathrm{cm}\cdot\mathrm{s}^2}]$	Absorption $[\frac{\mathrm{dB}}{\mathrm{MHz}\cdot\mathrm{cm}}]$
Fett	1470	0.97	$1.42 * 10^5$	0.5
Muskeln	1568	1.04	$1.63 * 10^5$	2
Leber	1540	1.055	$1.66 * 10^5$	0.7
Gehirn	1530	1.02	$1.56 * 10^5$	1
Knochen	3600	1.7	$6.12 * 10^5$	4-10
Wasser	1492	0.9982	$1.49 * 10^5$	0.002

Tabelle 2.1: Typische akustische Gewebeparameter; Tabelle entnommen aus (Mor95)

2.2.3 Reflexion und Brechung

Die Region zwischen zwei aneinandergrenzenden Medien wird im Folgenden als Grenzschicht bezeichnet. Trifft die Schallwelle auf eine solche Grenzschicht, so wird sie abhängig vom Schallimpedanzunterschied der beiden Medien gebrochen und eventuell reflektiert. Dieser Vorgang ist in Abbildung 2.1 schematisch dargestellt. Der Einfallswinkel ist gleich dem Reflexionswinkel. Der Brechungswinkel ist von der Impedanzänderung abhängig. Zur Berechnung des Brechungswinkels wird das Snelliussche Brechungsgesetz verwendet,

$$\frac{\sin(\theta_2)}{\sin(\theta_1)} = \frac{Z_1}{Z_2} \tag{2.5}$$

mit θ_1 als Einfallswinkel, θ_2 als Brechungswinkel und Z_1, Z_2 als Schallimpedanzen der korrespondierenden Medien.

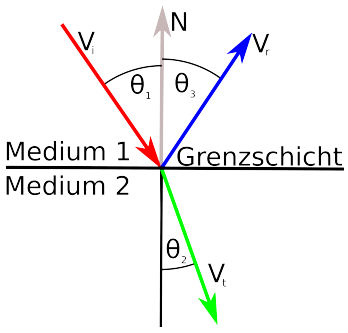


Abbildung 2.1: Reflexion und Brechung einer Schallwelle

Die Schallenergie der Welle teilt sich auf die beiden neu entstandenen Wellen auf. Das Verhältnis der einfallenden Welle zur reflektierten Welle ist durch den Reflexionskoeffizienten R gegeben. Das Verhältnis der einfallenden Welle zur gebrochenen Welle durch den Brechungskoeffizienten B. Zur Berechnung der

beiden Koeffizienten werden die Fresnelschen Formeln verwendet:

$$R = \left(\frac{Z_1 \cos\theta_1 - Z_2 \cos\theta_2}{Z_1 \cos\theta_1 + Z_2 \cos\theta_2} \right)^2 \tag{2.6}$$

$$B = \frac{4 Z_1 \cos\theta_2 Z_2 \cos\theta_1}{Z_1 \cos\theta_1 - Z_2 \cos\theta_2} \tag{2.7}$$

Wie an den beiden Formeln zu erkennen ist, nimmt der Reflexionskoeffizient R mit steigendem Einfallswinkel ab, während die Brechung B zunimmt. Je größer die Differenz der beiden Schallimpedanzen Z_1 und Z_2 ist, desto größer ist die Reflexion. Sind beide Schallimpedanzen identisch, dann findet keine Reflexion statt und der Brechungswinkel ist identisch mit dem Einfallswinkel.

2.2.4 Absorption

Eine sich in homogenem Gewebe ausbreitende Schallwelle wird auf Grund von Energieverlusten, verursacht durch innere Reibung, Wärmeleitung, Relaxation und Streuung, geschwächt. Die Absorption erfolgt exponentiell mit zunehmender Entfernung l vom Schallkopf. Sie ist abhängig vom Absorptionskoeffizienten α des durchlaufenden Mediums und der Frequenz f. Je höher die Frequenz, desto höher ist die Abschwächung. Um die verbleibende Intensität I einer Schallwelle mit einer Anfangsintensität von I_0 zu berechnen, wird Formel 2.8 verwendet.

$$I(l) = I_0 e^{-\alpha l f} \tag{2.8}$$

Für biologisches Gewebe lässt sich die Abhängigkeit von der Frequenz näherungsweise angeben. Die Schallwellenintensität verringert sich pro Zentimeter um ungefähr $1 \frac{\text{dB}}{\text{MHz}}$. Je weiter der Schall sich von der Quelle entfernt, desto höher ist seine Abschwächung, wodurch seine maximale Eindringtiefe begrenzt ist. Zur Untersuchung von weit entferntem Gewebe muss daher die Frequenz entsprechend angepasst werden. Da die Frequenz, wie in Abschnitt 2.3.2 zu sehen, einerseits die Auflösung und andererseits die Absorption bestimmt, ist es sinnvoll je nach gewünschter Eindringtiefe die höchstmögliche Frequenz zu verwenden.

Besteht die zu durchschallende Region aus inhomogenem Gewebe, so muss dies berücksichtigt werden. Jede inhomogene Region lässt sich in beliebig kleine homogene Regionen aufteilen. Um die gesamte Absorption für die durchschallte Region zu berechnen, werden alle Absorptionsanteile miteinander multipliziert.

2.2.5 Streuung

Eine Schallwelle, die durch ein Medium wandert, wird gestreut, falls es sich nicht um ein perfekt homogenes Medium handelt. Ein Teil der Schallenergie bewegt sich nicht mehr längs der ursprünglichen Wellenrichtung. Die Energie wird,

abhängig von der Art der Streuung, in unterschiedliche Richtungen reflektiert. Beim Ultraschall unterscheidet man verschiedene Streuungen, die nach Abhängigkeit des durchschnittlichen Durchmessers a der Partikel in dem durchschallten Medium eingeteilt werden. Es wird zwischen der geometrischen-, stochastischen- und der Rayleigh-Streuung unterschieden. In Tabelle 2.2 sind die verschiedenen Streuungen sowie die wichtigsten Eigenschaften der jeweiligen Streuart schematisch angeordnet.

Streuung	Relation	Frequenzabhängigkeit	Streustärke	Beispiel
geometrisch	$a \gg \lambda$	linear	stark	Gefäße
stochastisch	$a \sim \lambda$	gemischt	mittel	Leber
Rayleigh	$a \ll \lambda$	biquadratisch	schwach	Blut

Tabelle 2.2: Streubereiche für Ultraschall

Die geometrische Streuung tritt zwischen Grenzschichten mit stark abweichender Schallimpedanz auf und ist in Abschnitt 2.2.3 beschrieben. Bei Partikeln in der Größenordnung der Wellenlänge λ tritt stochastische Streuung auf, die als Miestreuung bezeichnet wird. Ein diffuser Streukegel - wie in Abbildung 2.2 zu sehen - wird zurückgestreut. Die Ausrichtung des Kegels ist abhängig vom Durchmesser a der Partikel. Zur Bestimmung der richtungsabhängigen Verteilung I_α, auch als Phasenfunktion bezeichnet, wird die Henyey-Greenstein-Phasenfunktion (2.9) verwendet (HG41).

$$I_\alpha = \frac{1 - g^2}{\sqrt{(1 + g^2 - 2g\cos(\alpha))^3}} \tag{2.9}$$

Der Asymmetriefaktor g gibt hierbei das Verhältnis zwischen der Wellenlänge λ und der Partikelgröße a an. Diese Streuung ist neben der Rayleigh-Streuung der Grund weshalb ein Echo registriert wird, wenn die Ultraschallwelle auf eine nicht senkrecht zum Strahl angeordnete Grenzfläche trifft.

Rayleigh-Streuung ist die Streuung an Partikeln, deren Durchmesser in Relation zur Wellenlänge sehr klein ist. Der Streukoeffizient $a_{R\lambda}$ ist definiert durch die Differenz der Intensität der einfallenden Welle I_0 zur Intensität der Welle nach der Streuung I, dividiert durch I_0. Er kann durch Formel 2.10

$$\frac{I_0 - I}{I_0} = a_{R\lambda} = \frac{8\pi^3 \left(n_\lambda^2 - 1\right)^2}{3\lambda^4 N} \tag{2.10}$$

mit n_λ als lichtwellenabhängigem Brechungsindex und N als Anzahl der Moleküle pro cm^3 berechnet werden.[3]

[3] Anmerkung: Die blaue Farbe des Himmels entsteht durch Rayleigh-Strahlung. Das Licht wird

Die winkelabhängige Verteilung des gestreuten Lichtes erfolgt nach der Beziehung (2.11),

$$I_\alpha = \frac{3\,(1 + \cos^2(\alpha))}{4} \tag{2.11}$$

wobei α den Winkel zwischen Einfallsrichtung und Emissionsrichtung angibt.

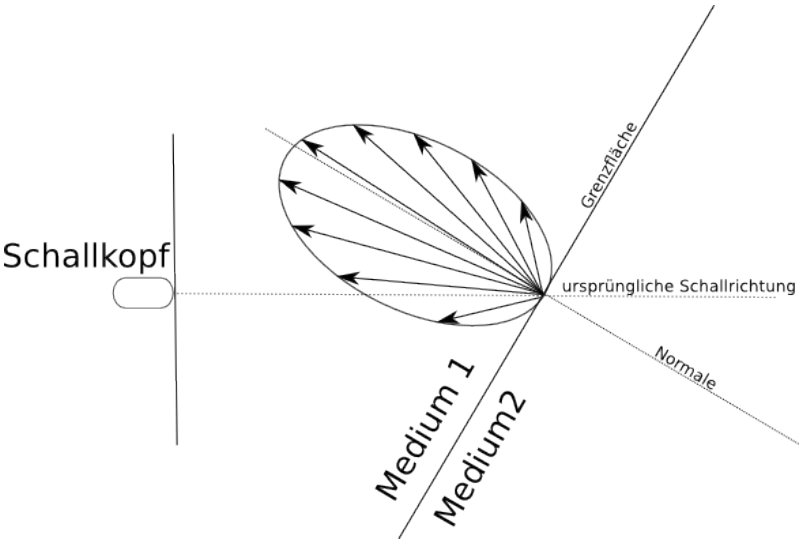


Abbildung 2.2: Reflektierter Streuungskegel an rauen Oberflächen

2.2.6 Piezoelektrischer Effekt

Der indirekte Piezoeffekt bezeichnet die Materialverformung von Festkörpern, wenn diese an eine elektrische Spannung angeschlossen werden. Durch eine schnelle elektrische Ansteuerung eines Piezoelementes kann somit ein Ultraschallimpuls erzeugt werden. Der Effekt funktioniert umgekehrt. Beim direkten Piezoeffekt entsteht eine elektrische Spannung an einem Festkörper, wenn dieser elastisch verformt wird. Das Piezoelement kann daher auch als Detektor für Schallwellen genutzt werden, indem es die empfangenen Schallwellen in elektrische Impulse umwandelt.

2.3 Bildgebung

2.3.1 Echo-Impuls-Verfahren

Alle Verfahren der bildgebenden Ultraschalldiagnostik arbeiten nach dem Echo-Impuls-Verfahren. In einem Ultraschallkopf wird mittels des piezoelektrischen

umso stärker gestreut, je kleiner die Wellenlänge ist. Daher wird blaues Licht stärker gestreut als zum Beispiel rotes Licht.

Effektes ein Ultraschallimpuls erzeugt. Dieser Schallimpuls breitet sich im Gewebe aus und baut ein Schallfeld auf. Aufgrund der im vorherigen Abschnitt beschriebenen Phänomene Streuung und Reflexion entstehen Echos, die zurück zum Schallkopf geworfen werden. Dieser wandelt die empfangenen Signale in elektrische Impulse um. Durch eine Auswertung der Stärke dieser Impulse sowie deren Laufzeit, ist es möglich eine Abbildung des durchschallten Objekts zu erzeugen. Unter der Annahme einer konstanten Schallgeschwindigkeit c, kann die Entfernung l des Objektes zum Schallkopf berechnet werden.

$$l = \frac{c\,(t_{received} - t_{send})}{2} \tag{2.12}$$

Dabei bezeichnet $t_{received}$ den Zeitpunkt an dem das Signal empfangen wurde und t_{send} den Zeitpunkt der Schallerzeugung.

2.3.2 Aufnahmezeit

Wie im vorigen Abschnitt beschrieben, lässt sich die Entfernung l eines Objektes zum Schallkopf mit Formel 2.12 berechnen.

Bevor der Empfang von Echos möglich ist, müssen erst alle Echos des vorherigen Sendeimpulses abgeklungen sein. Ansonsten kann nicht unterschieden werden, von welchem Sendeimpuls die empfangenen Echos stammen. Daher ist die maximale Wiederholungsrate von der maximalen Eindringtiefe l_{max} und der Schallgeschwindigkeit c abhängig. Die Schallgeschwindigkeit in den durchschallten Bereichen ist i.d.R. nicht bekannt. Daher wird eine durchschnittliche Schallgeschwindigkeit von $1540\,\frac{\mathrm{m}}{\mathrm{s}}$ angenommen. Die maximale zeitliche Auflösung t_{max} lässt sich dann in Abhängigkeit der maximalen Eindringtiefe l_{max} mit Formel 2.13 berechnen.

$$t_{max} = \frac{2 \cdot l_{max}}{1540} \tag{2.13}$$

Die maximale Eindringtiefe hängt von der Anfangsstärke der Schallwelle, der Absorption im Medium und der Empfindlichkeit des Empfängers ab. Da die Absorption, wie in Abschnitt 2.2.4 beschrieben, von der Frequenz abhängt, ist die Eindringtiefe u. a. durch die Frequenz festgelegt. Es gilt: Je höher die Frequenz, umso geringer die Eindringtiefe.

Die Frequenz bestimmt, wie im folgenden Abschnitt erklärt, auch die räumliche Auflösung. Ein guter Kompromiss zwischen einer hohen Eindringtiefe und einer hohen Auflösung ist es, die höchstmögliche Frequenz für eine gegebene Eindringtiefe zu verwenden.

2.3.3 Schallfeld

Das von einem diagnostischen Schallkopf ausgehende Schallfeld ist in Abbildung 2.3 schematisch dargestellt. Es ist in zwei Bereiche eingeteilt. Unmittelbar hinter dem Schallkopf befindet sich das Nahfeld, das durch starke Interferenzerscheinungen geprägt ist. Danach schließt sich das Fernfeld an, das eine sich kontinuierlich ausweitende Strahlenkeule bildet. Zwischen diesen Bereichen befindet sich der Fokusbereich in dem die Schallintensität gebündelt ist.

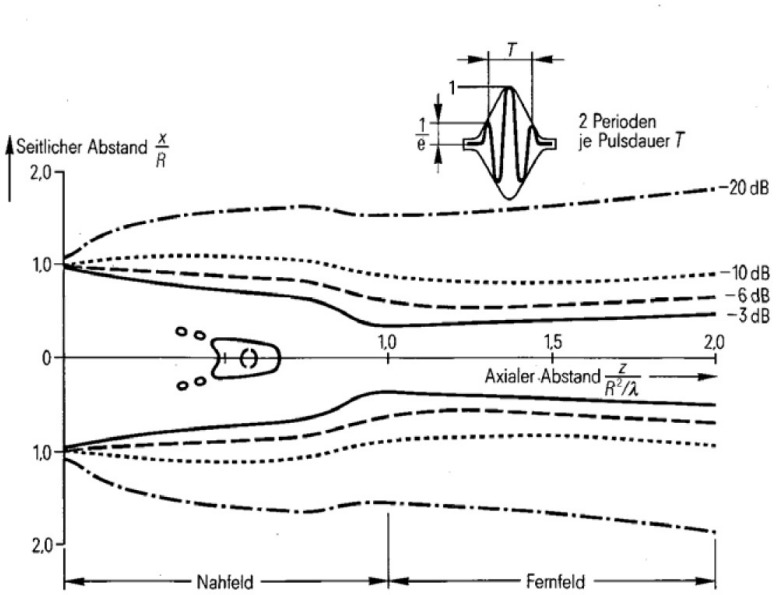


Abbildung 2.3: Schallfeld einer gepulsten Anregung für einen runden, ebenen Wandler. Abbildung entnommen aus (Mor95)

Als räumliche Auflösung wird der kleinstmögliche Abstand zweier Objekte, die gerade noch voneinander unterschieden werden können, bezeichnet. Wie zu sehen, ist sie im Fokusbereich am höchsten und verschlechtert sich mit zunehmendem Abstand zum Fokus. Sie lässt sich in die drei räumlichen Ausbreitungsrichtungen unterteilen.

Auflösungsvermögen

Die axiale Auflösung definiert die Auflösung in Ausbreitungsrichtung der Schallwellen. Sie wird direkt durch die verwendete Frequenz und die Schallgeschwindigkeit im Gewebe beeinflusst. Dies ist der Fall, da die Reflexion einer Schallwelle an einer Grenzfläche vollständig abgeschlossen sein muss, bevor die von einer dahinterliegenden Grenzfläche reflektierte Schallwelle erstere erreicht. Anderenfalls kann sie von letzterer nicht unterschieden werden.

Die laterale Auflösung definiert die Auflösung quer zur Ausbreitungsrichtung. Im Fokusbereich hängt die laterale Auflösung vom Durchmesser D des verwendeten Ultraschallwandlers ab. Näherungsweise beträgt die laterale Auflösung ein Drittel des Wandlerelementes.

Die elevationale Auflösung beschreibt die Auflösung senkrecht zur axialen und senkrecht zur lateralen Auflösung. Sie wird auch häufig als Schichtdicke bezeichnet und hängt von der Anordnung der Ultraschallwandler der verwendeten Fokussierungsmethode ab (Jen00). Die Schichtdicke wird häufig bei Ultraschallsimulationen und auch in der Ultraschallliteratur vernachlässigt. Bei Ultraschalluntersuchungen, in denen hohe Schallimpedanzunterschiede vorkommen, muss sie jedoch berücksichtigt werden (siehe hierzu auch Kapitel 2.3.6).

Ermittlung der Auflösung

Zwei gleichstarke Signalimpulse sind erst dann unterscheidbar, wenn sie mindestens um ihre Halbwertsbreite getrennt sind. Bei einem Grenzflächenabstand von mindestens $\lambda/2$ ist diese Bedingung erfüllt. Praktisch ist die Auflösung um mehr als den Faktor zwei schlechter. Dadurch ergibt sich Formel 2.14, mit der die bestmögliche axiale Auflösung Res_{ax} näherungsweise berechnet werden kann.

$$Res_{ax} = \frac{\lambda_{min}}{f} \tag{2.14}$$

λ_{min} steht für die minimale Schallgeschwindigkeit im Gewebe, die ca. 1460m/s beträgt und f für die Frequenz der Schallwelle.

Die laterale und elevationale Auflösung lassen sich über eine Punktbildfunktion (PSF engl. **P**oint-**S**pread-**F**unction) ermitteln. Ein punktförmiges Objekt wird in einer senkrechten Ebene durch den Strahl bewegt. Die Echointensität wird in Abhängigkeit des Ortes aufgetragen. Die Breite d, an der die Intensität um weniger als 6 dB geschwächt ist, bestimmt dann die Auflösung an der Stelle x. Das Verfahren wird in Abbildung 2.4 verdeutlicht. Für Simulationen wird häufig auch eine PSF zur Beschreibung des Schallfeldes an einem Punkt x verwendet. Diese ist i.d.R. eine Gaußfunktion, welche einfach zu beschreiben ist, eine hinreichende Genauigkeit besitzt und schnell zu berechnen ist. Kapitel 4.2.2 geht genauer auf diese Arten von Punktbildfunktionen ein.

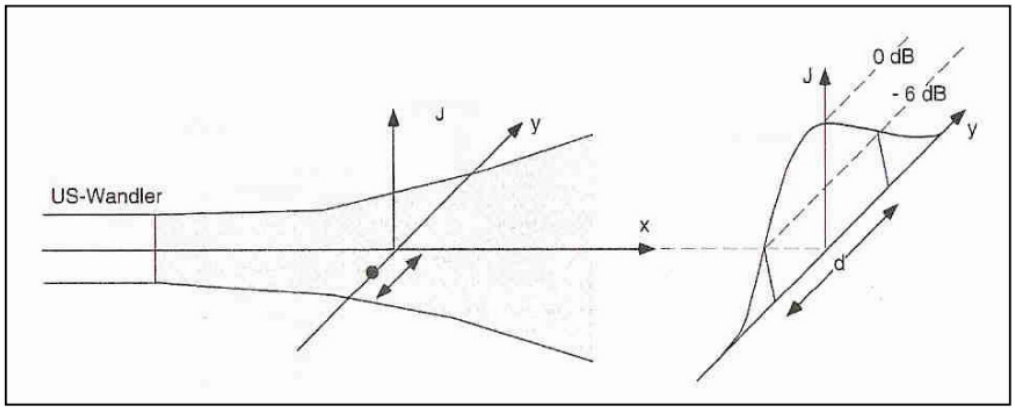


Abbildung 2.4: Verfahren zur Bestimmung der Auflösung über eine Punktbildfunktion. Abbildung entnommen aus (Dö8)

2.3.4 Darstellungsmethoden

Die gesammelten Ultraschalldaten können je nach Anwendungsgebiet auf unterschiedliche Weisen dargestellt und ausgewertet werden. In Abbildung 2.5 und 2.6 sind die unterschiedlichen Darstellungsmodi zu sehen.

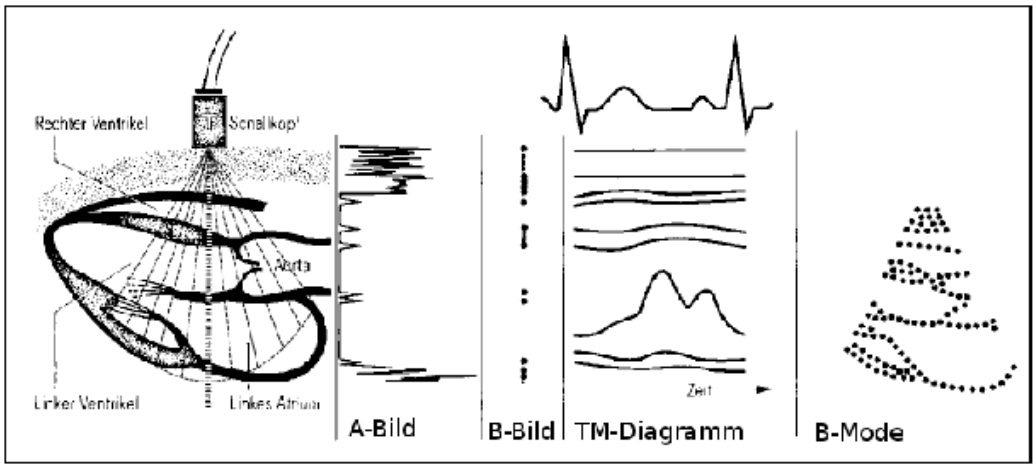


Abbildung 2.5: Skizze des durchschallten Herzens mit entsprechenden Ultraschallmodi. Abbildung entnommen aus (Mor95).

A-Bild

Die Amplitudenmodulation war die erste angewandte Darstellungsform der Sonographie. Das von dem Schallkopf empfangene Echo wird in einem Diagramm dargestellt. Die Intensität des empfangenen Echos wird auf die Ordinate, die

Verzögerungszeit, die der Tiefe entspricht, auf der Abszisse eingetragen. Je höher die Intensität ist, desto echogener ist das Gewebe in der angegebenen Tiefe. Der A-Mode liefert nur eindimensionale Daten und ist somit sehr schwer zu interpretieren. In der modernen Ultraschalldiagnostik findet er daher keine Anwendung mehr.

B-Bild

Die empfangenen Echoamplituden werden in diesem Modus als Graustufenwerte interpretiert. Das „B“ steht für engl. ***b****rightness*, also den Helligkeitswert des Bildes. Einem Echo mit niedriger Intensität wird ein dunkler Grauwert zugeordnet, einem Echo mit hoher Intensität ein heller Grauwert. Da das menschliche Auge nur 20 G bis 50 Graustufen gegeneinander abgrenzen kann, wird die Amplitude des empfangenen Echos von 120 dB auf 30 dB komprimiert. Das B-Bild ist somit einfach eine anders codierte Darstellung der A-Mode-Darstellung. Wie im A-Mode ist die Darstellung daher auch nur eindimensional.

M-Mode

Der M-Mode oder auch TM-Mode ist ein eindimensionales Verfahren, welches auch in der modernen Medizin noch häufig angewandt wird. Die Abkürzung steht für engl. *(****t****ime)****m****otion* also Bewegung. Die B-Bild-Grauwerte werden im Zeitverlauf dargestellt, indem die zeitlich aufeinander folgenden Echozeilen nebeneinander abgebildet werden. Bewegungen des Gewebes bzw. der untersuchten Strukturen haben Unterschiede in den einzelnen Impulsechos zur Folge. Bewegungsabläufe von Organen lassen sich hiermit eindimensional darstellen. Da die maximale Wiederholungsrate sehr hoch ist (mehrere kHz), eignet sich der M-Mode besonders gut für kardiologische Untersuchungen. Schnelle Bewegungen wie einzelne Bewegungen der Herzmuskelbereiche und der Herzklappen können hiermit genau untersucht werden.

B-Mode

Der am häufigsten verwendete Darstellungsmodus ist der B-Mode. Er liefert örtlich zweidimensionale Abbildungen. Das Schnittbild wird dabei aus einzelnen Linien zusammengesetzt, wobei für jede Linie eine Schallwelle ausgesendet und empfangen werden muss. Die Signalintensität wird dabei wie im B-Bild durch Graustufenwerte dargestellt. Eine Farbdarstellung wäre zwar technisch machbar und sinnvoll, da eine feinere Abstufung der empfangenen Amplituden möglich wäre, jedoch wird aus historischen Gründen in der Diagnostik die Grauwertdarstellung genutzt. Die Form des erzeugten Bildes hängt dabei vom eingesetzten Schallkopf ab. Dies wird in Abschnitt 2.3.5 genauer erklärt. Wird das Bild in Echtzeit generiert und dargestellt, so wird auch häufig die Bezeichnung 2D-Echtzeitmodus verwendet.

Doppler-Verfahren

Die Doppler-Sonographie ist eine Methode zur Messung der Geschwindigkeit des Blutes in Gefäßen. Das Verfahren nutzt den Effekt, dass sich die Frequenz der Schallwellen ändert, wenn sich Schallsender und Schallempfänger relativ zueinander bewegen. Diese als Dopplerfrequenz f_d bezeichnete Differenz tritt ebenso an bewegten Reflexionsflächen auf. Sie kann zur Berechnung der Blutflussgeschwindigkeit mittels der Formel:

$$f_d = 2 \cdot f_0 \cdot v \cdot cos(\phi/c) \tag{2.15}$$

genutzt werden. Die Geschwindigkeit v des Blutes ergibt sich dann unter Kenntnis der folgenden Variablen:

- der Schallgeschwindigkeit c in dem Medium
- der ausgesendeten Frequenz f_0
- dem Auftreffwinkel ϕ der Schallwelle im Verhältnis zum Blutfluss

Elastographie-Verfahren

Die Ultraschall-Elastographie ist eine Methode zur Visualisierung der viskoelastischen Eigenschaften des Gewebes. Mit dem Ultraschallkopf wird während der Untersuchung ein geringer Druck auf das Gewebe ausgeübt, wodurch sich dieses verformt. Die Größe der Verformungen werden dann in einem Dehnungsbild farblich kodiert dargestellt. Das Verfahren wird erst seit wenigen Jahren in der Ultraschalldiagnostik eingesetzt. Da Tumorgewebe eine unterschiedliche Elastizität im Vergleich zu normalem Gewebe besitzt, wird das Verfahren hauptsächlich zur Tumorerkennung verwendet.

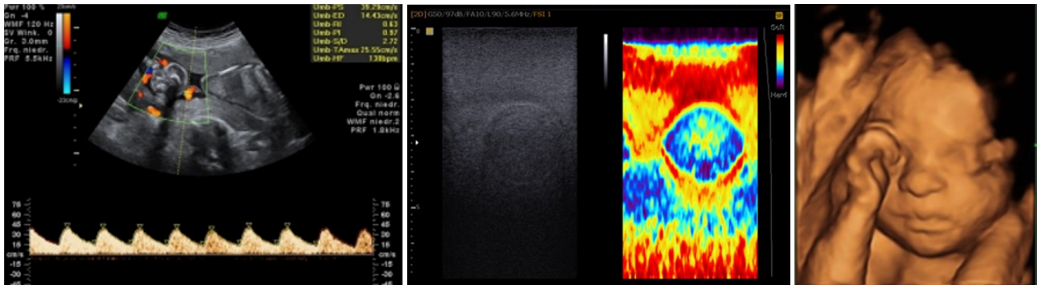


Abbildung 2.6: Links: Doppler-Verfahren. Mitte: Elastographie-Verfahren. Rechts: 3D-Ultraschall.

3D-Ultraschall

Als weitere Applikation wurde in den letzten Jahren die dreidimensionale Echographie entwickelt. Für ein dreidimensionales Bild wird zusätzlich zum Scan in einer Ebene ein Schwenk der Ebene vollzogen. Werden die Bilder dabei in Echtzeit generiert, handelt es sich um 4D-Ultraschall. Beide Verfahren werden häufig in der pränatalen Diagnostik eingesetzt, damit sich die Ärzte einen besseren visuellen Eindruck von dem Fötus machen können. Insbesondere beim 4D-Ultraschall lassen sich Herzerkrankungen und andere Gendefekte besser erkennen als bei einer gewöhnlichen Ultraschallbildgebung.

2.3.5 Ultraschallköpfe

Ein Ultraschallkopf (engl. *transducer*) dient zum Senden und Empfangen von Schallwellen, welche mittels eines piezoelektrischen Effektes erzeugt werden. Der Schallkopf besteht aus vielen kleinen Ultraschallwandlern, deren typischer Aufbau in Abbildung 2.7 zu sehen ist.

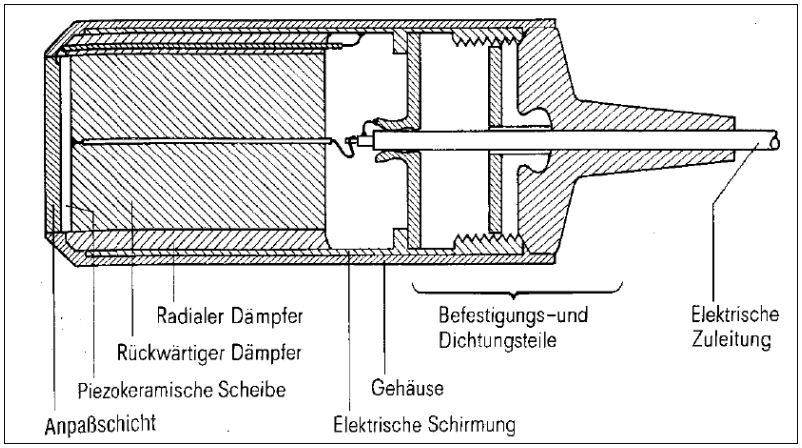


Abbildung 2.7: Aufbau eines Ultraschallwandlers, Abbildung entnommen aus (Mor95)

Für die Bildgebung werden Ultraschallköpfe mit 60 bis 400 solcher Wandler verwendet. Je nach Anordnung der Wandlerelemente werden die Schallköpfe nach den resultierenden Zeilenmustern benannt. Die verschiedenen Anordnungen, Linearscan, Konvexscan, Sektorscan und Kreisscan sind in Abbildung 2.8 skizziert.

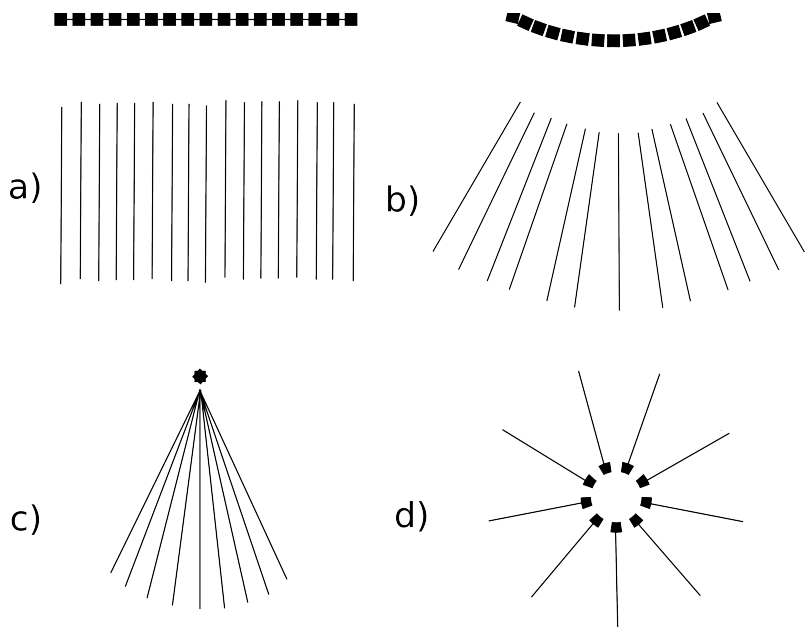


Abbildung 2.8: Verwendete Anordnungen von Ultraschallwandlern beim B-Scan: a) Linearscan, b) Konvexscan, c) Sektorscan, d) Kreisscan

Linearscan

Beim Linear-Array Schallkopf sind die Ultraschallwandler in einer Reihe angeordnet. Die Ansteuerung erfolgt zyklisch in Gruppen, wodurch eine lineare Scanbewegung entsteht. Die Linear-Array Schallköpfe werden hauptsächlich bei Standarduntersuchungen, wie zum Beispiel dem Abdominalultraschall eingesetzt. Die Elemente eines Konvex-Array Schallkopfes sind entlang eines Kreisbogens angeordnet, wodurch eine fächerförmige Abtastung entsteht. Dieser Schallkopftyp wird hauptsächlich bei transrektalen oder vaginalen Ultraschalluntersuchungen verwendet, aber auch bei Standarduntersuchungen wird er häufig eingesetzt.

Konvexscan und Sektorscan

Beim Sektorscan wird der Schallbereich elektromechanisch oder elektronisch verändert. Diese Art von Schallköpfen bietet den Vorteil einer kleinen Ankoppelungsfläche. Der Nachteil ist die schlechte Bildauflösung im Nahbereich des Schallkopfes. Bei der elektromechanischen Variante bewegt ein kleiner Elektromotor die Ultraschallwandler auf einem kleinen Kreisbogen. Heutzutage wird diese fächerförmige Abtastung fast ausschließlich durch eine zeitlich verzögerte Ansteuerung der Elemente erreicht. Diese Phased-Array Schallköpfe können sehr klein gebaut werden und ermöglichen dadurch die Untersuchung von Regionen auf kleinstem Raum.

Kreisscan

Beim Kreisscanner sind die Ultraschallwandler kreisförmig angeordnet oder es wird ein Ultraschallwandler analog zum elektromechanischen Sektorscan um 360 Grad rotiert. Dadurch entsteht ein komplettes 360-Grad Bild der Umgebung. Diese Ultraschallköpfe werden beim intravaskulären Ultraschall (IVUS) genutzt, um das Innere von Gefäßen zu untersuchen.

Fokussierung

Die Schallwellen können über verschiedene Methoden fokussiert werden. Die Ultraschallwandler können gekrümmt angeordnet werden oder es kann eine akustische Linse verwendet werden. Bei Phased-Array Schallköpfen erfolgt die Fokussierung durch die zeitlich verzögerte Ansteuerung der einzelnen Elemente. Durch die Fokussierung wird eine bessere räumliche Auflösung des Ultraschallsystems erreicht.

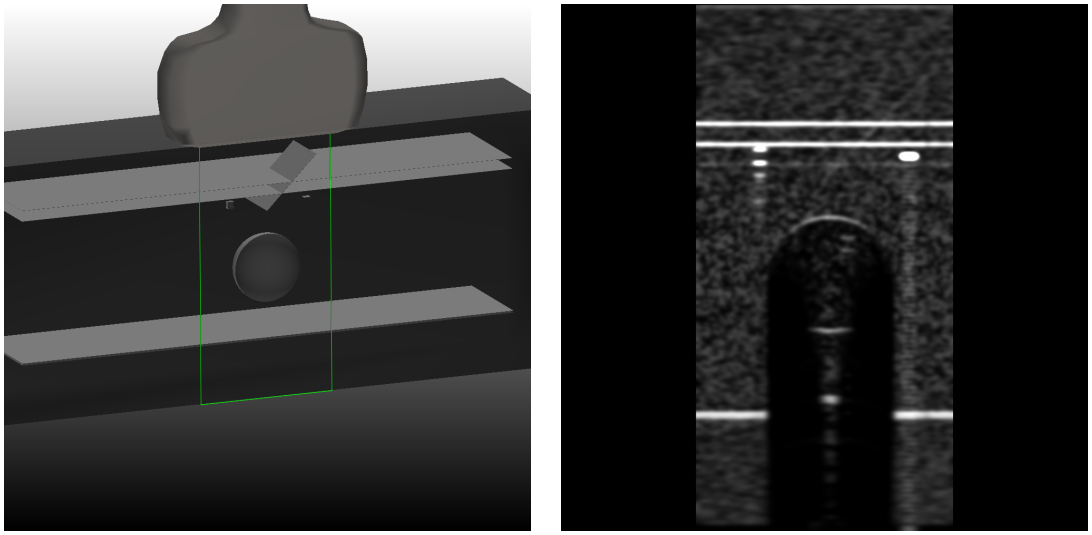
2.3.6 Artefakte

Die Sonographie visualisiert die akustischen Eigenschaften des durchschallten Mediums, indem es die Laufzeit des Schallimpulses misst. Da aber weder die exakten akustischen Eigenschaften noch die Länge des Mediums bekannt sind, gibt es mehr unbekannte als bekannte Variablen. Das Gleichungssystem ist daher unterbestimmt und es müssen folgende Annahmen getroffen werden, um das Ultraschallbild darstellen zu können:

- US breitet sich im gesamten Untersuchungsbereich mit einer konstanten Schallgeschwindigkeit c_{us} 1540 $\frac{\mathrm{m}}{\mathrm{s}}$ aus.
- Der Schallimpuls wird als unendlich dünner Ultraschallstrahl betrachtet.
- Die Ausbreitung des Ultraschallstrahles erfolgt geradlinig ohne Brechungen.
- Die Amplituden des empfangenen Echos sind proportional zum Impedanzunterschied nebeneinanderliegender Gewebeschichten.
- Die zurückgestreute Energie nimmt exponentiell mit zunehmender Laufzeit des Ultraschallstrahles ab.

Werden diese Annahmen verletzt, so werden Strukturen am falschen Ort dargestellt, sind unvollständig, haben die falsche Form und Größe oder die falsche Helligkeit. Diese falschen Bilddarstellungen werden als Ultraschallartefakte bezeichnet. Häufig lassen sich durch Artefakte auch Rückschlüsse auf die Eigenschaften des untersuchten Gebietes ziehen. Im Folgenden werden die auftretenden Artefakte

genauer beschrieben und dargestellt. Hierfür wird das in Abbildung 2.9a gezeigte Szenarium verwendet.



(a) (b)

Abbildung 2.9: Linke Seite: Komplettes Szenarium mit allen vorkommenden Objekten, rechte Seite: korrespondierendes Ultraschallbild.

In dem Szenarium werden verschiedene Objekte verwendet:

- (A) zwei große flache parallele Quader (oben)
- (B) ein flacher schräger Quader (oben)
- (C) ein kleiner Würfel (links, mitte)
- (D) eine große Kugel (mitte)
- (E) ein flacher großer Quader (unten)
- (F) zwei kleine flache parallele Quader (rechts, mitte)
- (G) ein großer Quader (enthält alle anderen Objekte)

Abbildung 2.9b zeigt das Szenarium mit realistischen Simulationsparametern und mehreren Artefakten. Die Simulationsparameter wurden so gewählt, dass die getroffenen Annahmen bestmöglich erfüllt sind. Um die jeweiligen Artefakte besonders hervorzuheben, wurden für die folgenden Simulationsbilder, im Vergleich zum Ausgangsszenarium, zwei Modifikation gemacht. Die Objekte (A-F) wurden deaktiviert, falls sie für die bessere Darstellung des Artefaktes nicht notwendig waren. Die Simulationsparameter wurden so abgeändert, dass die einzelnen Objekte sehr

gut zu erkennen sind. Hierfür wurde auch teilweise das in Ultraschallbildern übliche Speckle-Rauschen mit einem konstanten Grauton ersetzt. Die Abbildungen auf der jeweilig linken Seite zeigen das Ultraschallbild mit dem entsprechenden Artefakt. Die Abbildungen auf der jeweilig rechten Seite zeigen wie das Ultraschallbild ohne Artefakt aussehen würde.

Folgende Artefakte treten in realen Untersuchungssituationen auf:

Schichtdickenartefakt

Ein Ultraschallimpuls hat eine räumliche Ausdehnung, die man auch als Schallkeule bezeichnet (siehe Abschnitt 2.3.3). Trifft diese Schallkeule auf eine schräge Grenzfläche zwischen zwei Objekten mit unterschiedlicher Schallimpedanz, dann wird die Grenzfläche dicker und unschärfer im Ultraschallbild abgebildet.

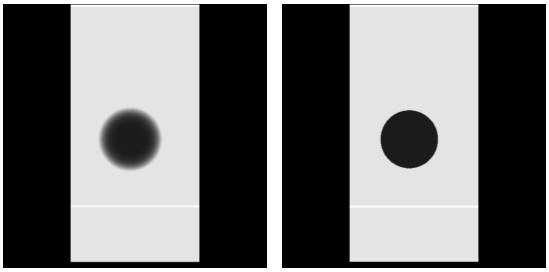


Abbildung 2.10: Die Objekte (A-C,F) sind deaktiviert und die Parameter von Objekt (D) und (G) wurden vertauscht.

Laufzeitartefakte

Hat ein geschalltes Objekt eine andere Schallgeschwindigkeit als die angenommene c_{us}, dann wird die Geometrie im Ultraschallbild verzerrt abgebildet. Bei einer höheren Schallgeschwindigkeit durchläuft der Schallimpuls das Objekt schneller als angenommen und das Objekt wird verkleinert im Ultraschallbild dargestellt. Analog hierzu werden Objekte mit einer kleiner Schallgeschwindigkeit als c_{us} vergrößert dargestellt. Objekte, die danach von dem gleichen Schallimpuls durchlaufen werden erscheinen ebenfalls verzerrt, auch wenn diese Objekte die angenommene Schallgeschwindigkeit c_{us} haben.

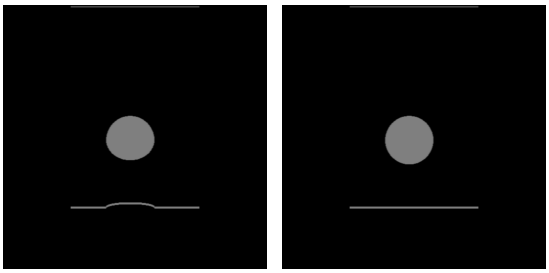


Abbildung 2.11: Die Objekte (A-C,F) sind in der Simulation deaktiviert.

Distale Abschattung

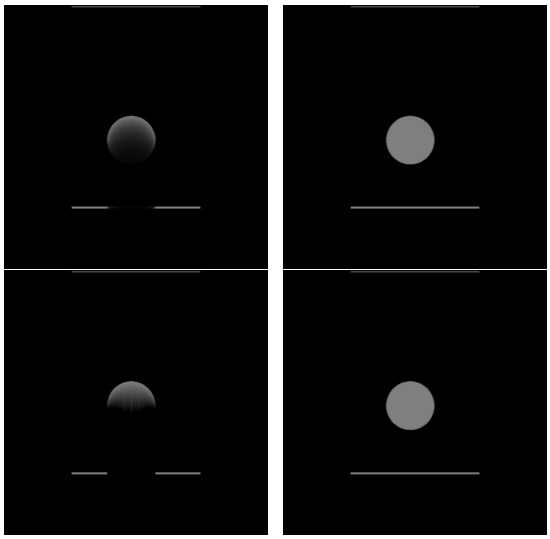


Abbildung 2.12: Die Objekte (A-C,F) sind in der Simulation deaktiviert.

Die distale Abschattung, auch Schallschatten genannt, kann durch zwei Varianten entstehen. Hat ein geschalltes Objekt eine höhere Schallabsorption als angenommen, so erscheint das Objekt mit zunehmender Distanz dunkler im Ultraschallbild. Objekte, die danach von dem gleichen Schallimpuls durchlaufen werden erscheinen ebenfalls dunkler, auch wenn diese Objekte die angenommene Schallabsorption haben. Haben zwei angrenzende Objekte eine hohe Schallimpedanzdifferenz zueinander, dann kann der Schall bei schrägem Auftreffwinkel zur Seite reflektiert werden. Die nachfolgenden Ultraschallreflektionen werden dann nicht mehr, oder nur noch sehr abgeschwächt, zum Schallkopf reflektiert. Sind dahinter liegende Objekte durch diesen Effekt auf dem Ultraschallbild nicht mehr sichtbar, dann spricht man auch von Schallauslöschung. Wird ein kugelförmiges Objekt in der Mitte getroffen, dann werden nur die äußeren Schallimpulse zur Seite reflektiert. Daher erscheint der Bereich unterhalb der Kugelränder dunkler im Ultraschallbild, weshalb dieses Artefakt Randschattenartefakt genannt wird.

Schallverstärkung

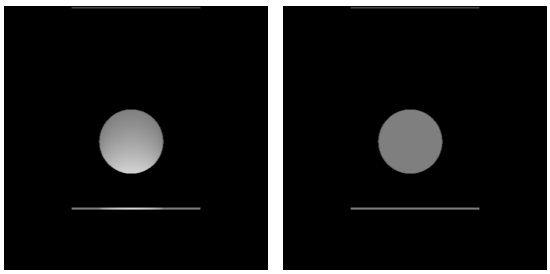


Abbildung 2.13: Die Objekte (A-C,F) sind in der Simulation deaktiviert.

Dieses Artefakt ist das Pendant zur ersten Variante des Schallschattens. Hat ein geschalltes Objekt eine niedrigere Schallabsorption als angenommen, dann erscheint es mit zunehmender Distanz heller im Ultraschallbild. Objekte, die danach von dem gleichen Schallimpuls durchlaufen werden, erscheinen ebenfalls heller, auch wenn diese Objekte die angenommene Schallabsorption haben.

Wiederholungsartefakt

Werden zwei annähernd parallele Grenzflächen mit starken Impedanzunterschieden geschallt, dann können Wiederholungsechos auftreten. Der Schallimpuls wird zwischen den Grenzflächen immer wieder reflektiert, wodurch es zu einer mehrfachen Darstellung der Grenzflächen im Ultraschallbild kommen kann.

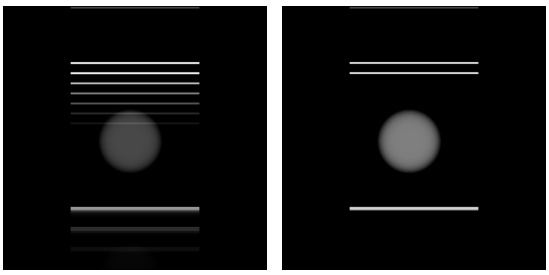


Abbildung 2.14: Die Objekte (B,C,F) sind in der Simulation deaktiviert.

Kometenschweifartefakt

Das Kometenschweifartefakt ist eine spezielle Form des Wiederholungsartefaktes. Die parallelen Grenzflächen befinden sich in einem sehr kleinen Abstand zueinander. Dadurch sind die Darstellungen im Ultraschallbild sehr dicht hintereinander und der dahinterliegende Bereich ist im Ultraschallbild kaum zu erkennen.

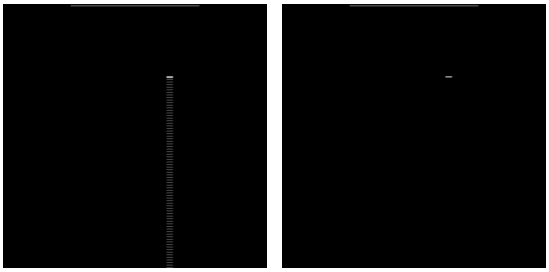


Abbildung 2.15: Die Objekte (A-E) sind in der Simulation deaktiviert.

Spiegelartefakt

Haben zwei angrenzende Objekte eine hohe Schallimpedanzdifferenz zueinander, dann kann der Schall bei bestimmten Auftreffwinkeln zur Seite reflektiert werden. Trifft der Schallimpuls nun wieder auf einen starken Reflektor, wird der Schallimpuls unter Umständen zurück zum Schallkopf reflektiert. Dieses empfangene Echo erscheint im Ultraschallbild hinter dem ersten Reflektor, obwohl das Objekt nicht in der ursprünglichen Schallimpulsrichtung liegt.

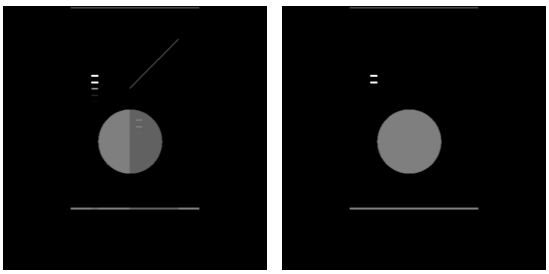


Abbildung 2.16: Die Objekte (A,F) sind in der Simulation deaktiviert.

Speckle-Rauschen

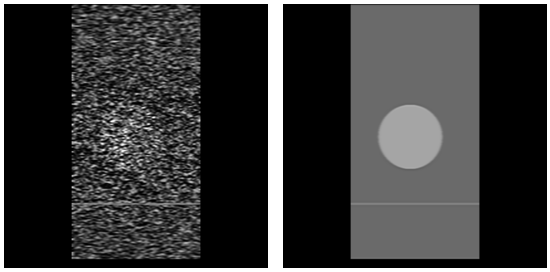


Abbildung 2.17: Die Objekte (A-C,F) sind in der Simulation deaktiviert.

Ein Objekt ist im Allgemeinen nicht homogen, sondern besteht aus vielen unterschiedlich großen Zellen oder Gewebestrukturen. Diese reagieren unterschiedlich stark mit dem ausgesendeten Schallimpuls. Dadurch entstehen charakteristische Interferenzmuster. Diese erscheinen im Ultraschallbild als viele dicht nebeneinanderliegende dunkle und helle Bildpunkte. Die Größe und Form dieser Speckles hängt stark von dem Aufbau des Ultraschallgerätes und der gewählten Frequenz ab. Das normale Bildrauschen ist bei den meisten Ultraschallbildern nicht mehr sichtbar, da das Signal über mehrere Bilder gemittelt wird.

3 Stand der Forschung

3.1 Simulationssysteme

3.1.1 IVUS

Die zu Schulungszwecken genutzten IVUS-Simulatoren lassen sich in zwei Kategorien einteilen. Die Simulatoren der ersten Kategorie verwenden physische Modelle, um die Handhabung der Geräte zu vermitteln. Die Simulatoren der zweiten Kategorie verwenden echte IVUS-Aufnahmen, um die korrekte Interpretation der Bilder zu vermitteln. Ein Simulator, der sowohl die Führung der Instrumente als auch die Interpretation der Bilder vermittelt, ist derzeit noch nicht auf dem Markt. Es gibt Publikationen, welche IVUS-Bilder anhand von Computermodellen simulieren können (Ram05, ABHM07). Es handelt sich aber hierbei bisher nur um Prototypen, für die noch keine geeigneten Lehrmodelle zur Verfügung stehen. Da die bisher genutzten Methoden gut funktionieren und der Zusatznutzen eines kombinierten IVUS-Simulators eher gering ist, wurde bisher kein solcher Simulator für die Lehre entwickelt.

3.1.2 Brachytherapie

Die korrekte Durchführung einer Brachytherapie wird mit mechanischen Simulatoren oder direkt am Patienten vermittelt. Die mechanischen Simulatoren bestehen aus echten transrektalen Ultraschallköpfen (TRUS) und Nadeln sowie einer künstlichen Prostata. Die Handhabung der Geräte und das Verhalten der Nadeln ist dadurch sehr realistisch. Allerdings gibt es im Ultraschallbild nur sehr wenige Strukturen zu sehen. Außerdem muss das Prostataphantom nach mehrmaligem Gebrauch ausgetauscht werden, da es durch die Nadeln beschädigt wird. Des Weiteren können an dem Phantom keine speziellen Komplikationen geübt werden. Aktuell existiert nur ein auf Software basierter Prototyp zur Simulation einer Brachytherapie. Dieser wird jedoch noch nicht im Lehrbetrieb eingesetzt. Der Simulator verwendet echte 3D-Ultraschallvolumen, deformiert diese mit Hilfe eines FEM-Modells und interpoliert mit Hilfe des deformierten Volumens ein 2D-Ultraschallbild (GSMS12).

3.1.3 Sonographie

UltraSim

UltraSim von MedSim ist ein Ultraschall-Trainings-Simulator. Er besteht aus einer Puppe sowie einer Ultraschallattrappe, die an einen PC angeschlossen ist. Der Simulator ermöglicht die Simulation von B-Mode, Farb- und Spektraldopplerultraschall. Es stehen viele verschiedene Fallstudien zur Verfügung u. a. für Abdomen-, Gynäkologie-, Schwangerschafts-, Brust-, Gefäß-, Nacken- und Notfalluntersuchungen. Zur Darstellung der Ultraschallbilder werden echte Aufnahmen verwendet, die in Abhängigkeit zur Position der Schallkopfattrappe dargestellt werden.

U/S Mentor

Der *U/S Mentor* ist ein neuer Ultraschallsimulator der Firma Simbionix. Der Simulator besteht aus einer Puppe sowie einer an einen PC angeschlossenen Schallkopfattrappe. Das Trainingsmodul enthält ein Praxistraining für die Echokardiographie und zahlreiche Funktionen, um dem Anwender grundlegende psychomotorische Ultraschall-Fähigkeiten zu vermitteln. Für die Zukunft ist die Entwicklung weiterer Ultraschallmodalitäten geplant. Da der Simulator sehr neu auf dem Markt ist, konnten seine Fähigkeiten bisher leider noch nicht geprüft werden, jedoch sehen die gezeigten Bilder sehr realistisch aus. Details darüber, wie genau die Bilder erzeugt werden, wurden bisher nicht veröffentlicht.

Ultrasound Simulator

Der *Ultrasound Simulator* der Firma Schallware ist mit Modulen für Interne Medizin, Notfallmedizin, Kardiologie und Gynäkologie erhältlich. Er besteht aus einer Puppe sowie einer Schallkopfattrappe, die an einen PC angeschlossen ist. Die Ultraschallbildgenerierung basiert auf der Interpolation echter 3D- bzw. 4D-Ultraschallvolumen. Der Simulator wird international vertrieben und wird in Deutschland für die Ausbildung von Ärzten an mehreren Standorten verwendet.

SonoSim-II

Der *SonoSim-II* ist ein Ultraschallsimulator der Firma Sonofit und ist aus den beiden Vorgängerprojekten *SonoSim-I* und *SonoTrainer* entstanden. Die Ultraschallbildgenerierung basiert auf der Interpolation echter 3D-Ultraschallvolumen. Es gibt zahlreiche Module für Untersuchungen wie Geburtshilfe, TEE, endovaginale Untersuchung, Kardiologie, Farbdoppler, TRUS, Gynäkologie und Mamma-Diagnostik. Des Weiteren existiert ein Punktionsmodul, welches die Simulation von Nadeln erlaubt. Die visuelle Qualität dieser Nadelsimulation ist allerdings eher mäßig, da für die Darstellung der virtuellen Nadel im interpolierten Ultraschallbild eine einfache Bildüberlagerungstechnik (Blending) verwendet wird.

SonoSimulator

Der *SonoSimulator* der Firma SonoSim besteht aus einem Laptop mit einer angeschlossenen Schallkopfattrappe. Die Ultraschallbilder werden wie bei den meisten anderen Simulatoren anhand von echten 3D-Ultraschallvolumen interpoliert. Zusätzlich zum Ultraschallbild kann noch ein virtueller menschlicher Körper angezeigt werden, um die Anatomie an der geschallten Position darzustellen. Der Simulator besitzt auch ein Punktionsmodul, das ebenso wie *SonoSim-II* eine virtuelle Nadel im interpolierten Bild anzeigt. Deformationen oder Ultraschallartefakte durch die Nadel werden auch hier nicht simuliert.

Nicht mehr weiterentwickelte Prototypen

Die folgenden Prototypen von Ultraschallsimulatoren basieren alle auf 3D-Ultraschalldatensätzen, die interpoliert werden, um die zugehörigen 3D-Ultraschallbilder zu erzeugen. Die Simulatoren wurden bereits seit mehreren Jahren nicht mehr weiterentwickelt. Der Vollständigkeit halber werden sie an dieser Stelle inklusive ihrer letzten erschienenen Publikation erwähnt:

- *UltraTrainer* vom Fraunhofer Institut - 1996 (SM98)
- *SONOSim3D* von der Universität Stralsund - 1998 (Ehr98)
- *Echographic simulation* von TIMC Laboratory - 2000 (THL+00)

3.1.4 Ultraschallsimulationen

Im Folgenden werden Programme, Bibliotheken oder Toolboxen vorgestellt mit denen man Ultraschall simulieren kann.

3.1.5 Field II

Die wohl am häufigsten genutzte Open-Source-Software zur Simulation von Ultraschall ist *Field II*. Es wird seit vielen Jahren von Jørgen Arendt Jensen entwickelt und ist unter vielen Betriebssystemen einsetzbar, da es auf Matlab basiert. Dadurch ist es sehr leicht erweiterbar und kann gut an die individuellen Bedürfnisse angepasst werden. Es verwendet die Tupholme-Stepanishen Methode, um die Ultraschallfelder zu berechnen. Es können Schallkopffelder und Ultraschallbilder simuliert werden. Die Simulation eines Ultraschallbildes benötigt mehrere Stunden auf einem Computercluster, weshalb das Programm nicht für eine Echtzeitsimulation geeignet ist.

3.1.6 k-Wave

Eine gute Alternative zu Field II ist die im Jahr 2012 neu in Matlab entwickelte Open-Source *„acoustic toolbox“ k-Wave*. Die von Bradley Treeby und Ben Cox (TJRC12) entwickelte Toolbox kann lineare und nichtlineare Wellenausbreitungen in 1D, 2D und 3D simulieren. Die Simulation basiert auf einer generalisierten Form der Westervelt-Gleichung, die mit einer k-space pseudospektralen Methode gelöst wird. Die Methoden sind in C++ implementiert und können teilweise auf der GPU ausgeführt werden, um die Berechnungen zu beschleunigen. Dadurch kann ein Ultraschallbild in weniger als fünf Stunden berechnet werden.

3.1.7 Ultrasim

Ultrasim ist eine frei erhältliche auf Matlab basierte Toolbox zur Simulation von Schallfeldern. Es berechnet die Schallfelder, indem das Rayleigh Integral gelöst wird(Hol01). Die Toolbox wird hauptsächlich zur Entwicklung von Schallköpfen genutzt. Ultraschallbilder können ohne größere Modifikationen nicht damit berechnet werden.

3.1.8 Wave2000, Wave2500 und Wave3000

Die *Wave Programme* von CyberLogic sind Programme zur Simulation von Ultraschall. Sie werden hauptsächlich für die Entwicklung von Schallköpfen verwendet, können jedoch auch zur Simulation von Ultraschallbildern verwendet werden (KS03). Die Software berechnet die Lösung zur viskoelastischen Wellengleichung. Über die Geschwindigkeit der Simulation werden keine Angaben gemacht. Die Programme laufen nur unter Windows und sind nur kommerziell als Jahreslizenzen erhältlich.

3.2 Verfahren zur Simulation von Ultraschall

Es existieren zwei verschiedene Ansätze zur Simulation von 2D-Ultraschallbildern. In den nächsten beiden Abschnitten werden die Funktionsweise und die Vor- und Nachteile des Schnittbildverfahrens sowie des generativen Verfahrens beschrieben.

3.2.1 Das Schnittbildverfahren

Das Schnittbildverfahren verwendet 3D-Ultraschallvolumen, die über eine Schnittebene gesampelt werden, um daraus 2D-Ultraschallbilder zu erzeugen. Die Erstellung der 3D-Volumen wird heutzutage mit einem 3D-Ultraschallkopf gemacht. Früher wurden aufwendige Rekonstruktionsverfahren verwendet, um mehrere 2D-Ultraschallbilder zu einem 3D-Ultraschallbild zu kombinieren (ACO98). Um

aus dem 3D-Volumen ein 2D-Schnittbild zu erstellen, existieren verschiedene Verfahren[1]. Aufgrund der hohen Qualität und Ausführungsgeschwindigkeit wird i.d.R. ein trilineares Interpolationsverfahren verwendet, welches leicht mit Hilfe von 3D-Texture Mapping auf der Grafikkarte berechnet werden kann. Das so interpolierte Ultraschallbild kann sehr schnell in Echtzeit simuliert werden und die Implementierung ist sehr einfach. Die Bilder sehen realistisch aus, allerdings fehlen die typischen Ultraschallartefakte. Daher gibt es einige Ansätze diese Artefakte im Nachhinein künstlich in die erstellten Ultraschallbilder einzufügen. Dadurch kann zwar das Schallschattenartefakt bedingt simuliert werden, jedoch werden andere wichtige Artefakte wie Spiegel- und Mehrfachechos, Schallverstärkung, Reflexionen und Laufzeitartefakte nicht berücksichtigt (ACO98, THL+00, DLT+06, GS09). Die meisten in der Praxis genutzten Ultraschallsimulatoren nutzen das Interpolationsverfahren, da die Modellgenerierung einfach ist und die Bildqualität überzeugend aussieht.

3.2.2 Generative Verfahren

Diese Verfahren simulieren das komplette Ultraschallbild mit Hilfe von verschiedenen Verfahren. Die 3D-Modelle werden nur zur Beschreibung der Geometrie verwendet. Es können verschiedene Modelle verwendet werden wie CT, MRT oder Drahtgittermodelle. Die Modellgenerierung ist deutlich aufwendiger als bei dem Schnittbildverfahren, da die Modelle i.d.R., bevor sie für die Simulation genutzt werden können, noch bearbeitet werden müssen. Das generative Verfahren lässt sich in zwei weitere Gruppen unterteilen. Die exakten Ansätze versuchen die Simulation so genau wie möglich zu simulieren und die approximativen Ansätze versuchen die Simulation in Echtzeit zu berechnen.

Generative exakte Verfahren

Diese Verfahren versuchen die Schallwellen möglichst genau zu simulieren, indem sie versuchen die Wellengleichung

$$\left(\triangle - \frac{1}{c^2} \frac{\partial^2}{\partial t^2} \right) u = 0 \tag{3.1}$$

möglichst genau zu approximieren. Hierbei ist u die zu untersuchende Funktion und c die Schallgeschwindigkeit. Die Simulationen werden i.d.R. verwendet, um das Schallfeld von Schallköpfen zu optimieren. Für Echtzeitsimulationen sind sie nicht geeignet, da die benötigte Rechenzeit zu lang ist. So kann eine einzige Simulation auf einem Standard-PC mehrere Tage dauern. Da der Kern dieser Arbeit auf Echtzeitsimulation liegt, werden hier nur kurz die wichtigsten Publikationen beschrieben. 1992 wurde eine Lösung dieser Gleichung mit Hilfe der Green'schen

[1] in anderen Modalitäten wie CT oder PET wird dieses Verfahren auch als Reslicing bezeichnet.

Funktion und einer Finite Element-Methode formuliert (JS92). Die wohl gängigste Open-Source-Toolbox zur Simulation von Ultraschall ist *Field II* (Jen96). Die Matlab-Toolbox diskretisiert das Schallfeld durch mehrere hunderttausend Streupunkte. Die räumliche Impulsantwort eines Streupunktes $\overrightarrow{r_1}$ an einem diskreten Punkt $\overrightarrow{r_2}$ des Schallkopfes, zum Zeitpunkt t und der Schallgeschwindigkeit c wird durch das folgende Rayleigh-Integrals approximiert (Jen97):

$$h\left(\overrightarrow{r_1}, t\right) = \int_S \frac{\delta\left(t - \frac{|\overrightarrow{r_1} - \overrightarrow{r_2}|}{c}\right)}{2\pi|\overrightarrow{r_1} - \overrightarrow{r_2}|} dS. \tag{3.2}$$

Karamalis et al. verwenden die Westervelt Gleichung, um die Ausbreitung der Schallwellen zu modellieren (KWN10).

$$\nabla^2 p - \frac{1}{c_0^2}\frac{\partial^2 p}{\partial t^2} + \frac{\delta}{c_0^4}\frac{\partial^3 p}{\partial t^3} + \frac{\beta}{\rho_0 c_0^4}\frac{\partial^2 p^2}{\partial t^2} = 0 \tag{3.3}$$

Bei der Gleichung werden die thermale Abschwächung durch den Koeffizienten δ und die Nichtlinearität durch den Koeffizienten β mit einbezogen. Die anderen bereits bekannten Größen sind der Schalldruck p, die Schallgeschwindigkeit c und die spezifische Dichte ρ_0.

Dieser neue Ansatz und die Verwendung von performanten Programmiersprachen wie CUDA und C++ erlauben bei ähnlicher Genauigkeit eine deutlich schnellere Simulation als die Methode von *Field II*. Allerdings dauert die Simulation eines Bildes immer noch ca. eine Stunde und ist daher auch in naher Zukunft nicht echtzeitfähig.

Generative Echtzeitsimulation

Eine der ersten englischsprachigen Publikationen über die schnelle Simulation von Ultraschallbildern wurde im Jahr 2000 von Song et al. (SHS00) präsentiert. Die Simulation verwendet ein Drahtgittermodell (Mesh-Modell) und Strahlverfolgungs-Techniken (Ray-tracing), um Reflexionen an Gewebeübergängen zu simulieren. Speckle-Rauschen wird in den Bildern nicht simuliert und die Aufnahmequalität, der im Paper gezeigten Bilder ist leider zu schlecht, um eine eigene Aussage über den Realismus der Bilder machen zu können. Es werden keine Angaben über die Berechnungsdauer für ein Bild gemacht. Anhand eigener Untersuchungen wurde jedoch gezeigt, dass die Simulation, zumindest auf einem aktuellen PC, echtzeitfähig ist.

Laguitton et al. (LPD05) publizierten im Jahr 2005 eine Ultraschallsimulation basierend auf CT-Volumen. Sie extrahieren eine 2D-Schicht aus dem CT-Volumen und benutzen das Modell von Bamber und Dickinson (BD80), um Speckle-Rauschen zu dem Bild hinzuzufügen. Ultraschallartefakte wie Reflexionen oder Schallschatten fehlen komplett. Dies und eine zu fein simuliertes Speckle-Rauschen führen dazu,

dass das gezeigte Bild unrealistisch wirkt. Die Simulation erreicht noch keine Echtzeitwerte, da für die Berechnung eines Bildes fünf Sekunden benötigt werden.

Im gleichen Jahr entwickelten Schwenk et al. (Sch05) im Rahmen einer Diplomarbeit einen heuristischen Ansatz zur Generierung von künstlichen Ultraschallbildern. Das Verfahren basiert auf einer vereinfachten Strahlverfolgungs-Technik (Ray-casting) und kann in Echtzeit Ultraschallbilder berechnen. Diffuse Reflexionen werden mit einem gradientenbasierten Verfahren berechnet, welches zudem noch einen gewebeabhängigen Faktor sowie die Intensität des Schallimpulses berücksichtigt. Das Speckle-Muster wird mit sogenannten Speckle-Splats simuliert. Jeder Abtastpunkt besitzt mit einer bestimmten Wahrscheinlichkeit einen Speckle-Splat mit einer wahrscheinlichkeitsbestimmten Intensität. Die Summe aller überlagerten Speckle-Splats an einem Abtastpunkt gibt hierbei seine Intensität an. Die Schichtdicke kann durch eine gewichtete Überlagerung von mehreren parallelen Schichten simuliert werden. Allerdings ist in diesem Fall die elevationale und laterale Auflösung konstant. Der Fokusbereich des Schallfeldes erstreckt sich also über das komplette Ultraschallsignal. Die erzeugten Bilder sehen optisch gut aus und die Qualität der Schallschatten kann überzeugen. Allerdings werden bei dem Ansatz unterschiedliche Schallgeschwindigkeiten der Medien nicht berücksichtigt, wodurch viele Artefakte nicht simuliert werden können.

Ein Jahr später entwickelten Zhu et al. (ZMRK06) ein Simulationssystem zur ultraschallgeführten Nadelplatzierung. Die Ultraschallsimulation basiert auf segmentierten CT-Daten und ist echtzeitfähig. Das typische Aussehen von Ultraschallbildern wird durch 3D-Texturen erzeugt, die mit Hilfe von echten Ultraschallbildern generiert werden. Um die Unschärfe von Ultraschallbildern zu simulieren, werden Bildnachbearbeitungstechniken wie „Alpha Blending“ verwendet. Zur Simulation der Schallschatten werden Ray-casting-Techniken eingesetzt.

Im Jahr 2007 wurden Ultraschallsimulationen von drei verschiedenen Gruppen entwickelt. Wein et al. (WKC+07) entwickelten eine echtzeitfähige Ultraschallsimulation zur Registrierung von 3D-Ultraschallvolumen und CT-Volumen. Als Ausgangsdaten dienen CT-Volumen, für welche eine Mappingfunktion verwendet wird, um die in der CT verwendeten Hounsfield-Einheiten als Schallimpedanzen zu interpretieren. Mit Hilfe der Schallimpedanzen werden die Schallreflexionen und die Schallintensitäten für alle Abtaststrahlen berechnet. Anschließend wird Perlin-Rauschen hinzugefügt, um das typische Speckle-Rauschen von Ultraschallbildern zu simulieren. Die Simulation ist in der Lage, distale Schallschatten zu simulieren, allerdings werden die Abschwächungskoeffizienten nur an Gewebeschichten ausgewertet, deshalb ist nur ein sehr abrupter Schallschatten möglich. Des Weiteren wird die Ultraschallgeschwindigkeit als konstant angenommen, wodurch alle Abtaststrahlen nur gerade verlaufen. Daher fehlen die übrigen typischen Ultraschallartefakte. Auch das typische Speckle-Muster von Ultraschallbildern ist nicht realistisch, da eine Verschlechterung der lateralen Auflösung mit zunehmender Distanz vom Schallkopf nicht berücksichtigt wird. Der Vorteil dieser Methode

ist, wie auch bei den interpolierenden Verfahren, die einfache Modellgenerierung. Die CT-Volumen müssen nicht weiter bearbeitet werden, es muss nur einmal die Mappingfunktion erstellt werden.

Vidal et al. (VJHG08) entwickelten eine Simulation zur ultraschallgeführten Nadeleinführung. Die zur Simulation genutzten Modelle sind CT-Volumen und 3D-Texturen. Diffuse Reflexionen werden mit dem kantenverstärkenden Sobelfilter simuliert und Schallschatten nachträglich in das Bild eingefügt. Der Schwerpunkt der Arbeit liegt mehr auf der Nadelsimulation, als auf der Simulation von Ultraschall.

Der Autor dieser Arbeit entwickelte eine Ray-tracing basierte echtzeitfähige Ultraschallsimulation auf Basis von segmentierten CT-Volumen im Rahmen einer Diplomarbeit (BAH07). Die Speckles wurden ähnlich wie in der vorliegenden Arbeit mit Hilfe einer 2D-Faltungsmethode erzeugt. Da viele Berechnungen wie Normalen und Distanzfelder vorberechnet werden, können Deformationen nur genutzt werden, wenn diese bereits vorberechnet sind. Auch die elevationale Ausdehnung des Schallpulses wird nicht berücksichtigt.

Shams et al. (SHN08) publizierten 2008 einen Simulator für klinische Ultraschallbilder, basierend auf einem Ray-casting von CT-Volumen. Die Simulation ist echtzeitfähig und berechnet die diffusen Reflexionen mit einem Lambertschen Rauschmodell. Die Schallimpedanzen für die Berechnung der diffusen Reflexionen werden mit einer manuell angepassten Version von Weins Mappingmethode berechnet (WKC+07). Das Speckle-Rauschen wird mit einem vorberechnetem Rauschvolumen (Scatter-Volume) simuliert, das durch *Field II* erzeugt wird. Zusätzlich ist die Breite des Schallimpulses und somit die laterale Auflösung ortsabhängig modulierbar. Die Abschwächung des Schallimpulses wird nur in Abhängigkeit der Tiefe und der Auftreffwinkel berechnet, ist allerdings nicht abhängig vom Medium. Daher lassen sich nur bestimmte Arten von Schallschatten simulieren, die anderen typischen Ultraschallartefakte sind mit der Methode nicht simulierbar.

Reichl et al. (RPAS09) entwickelten 2009 eine auf CUDA basierte echtzeitfähige Ultraschallsimulation. Die Methode basiert im Wesentlichen auf dem Verfahren von Wein et al. (WKC+07) mit einer modifizierten Mappingfunktion und verbesserter Berücksichtigung der Absorption.

Kutarnia et al. (KPY10) entwickelten 2010 die erste Ultraschallsimulation basierend auf MRI-Volumen. Die Methode verwendet *Field II* um ein Rauschvolumen zu erzeugen, die genaue Methode wird jedoch nicht beschrieben. Dieses Volumen wird dann, wie bei den interpolierenden Verfahren, interpoliert, um ein 2D-Bild zu erhalten. Die Qualität der simulierten Bilder ist nicht für einen medizinischen Simulator geeignet, da alle typischen Artefakte fehlen.

Sun et al. (SM11) veröffentlichen ein Jahr später eine echtzeitfähige Sonographiesimulation für medizinisches Training. Die Simulation verwendet eine 2D-Schicht, die aus einem 3D-Mesh-Modell erzeugt wird. Dies hat den Vorteil

gegenüber anderen Methoden, dass das 3D-Mesh-Modell sehr leicht deformiert werden kann, um zum Beispiel ein schlagendes Herz zu simulieren. Das charakteristische Aussehen von Ultraschallbildern wird mit selbst erzeugten Texturen simuliert. Ultraschalltypische Artefakte können mit der Simulationsmethode nicht erzeugt werden.

Law et al. (LUK+11) verwenden auch ein 3D-Mesh-Modell zur Simulation von Ultraschall. Die Methode basiert auf einem Ray-tracing-Verfahren, wobei das englische Wort „Tracing“ in diesem Fall irreführend ist. Es handelt sich vielmehr um ein Ray-casting-Verfahren, da die Schallgeschwindigkeit in dem Paper keine Rolle spielt und die Brechung von Strahlen nicht berücksichtigt wird. Die Simulation der Bilder basiert auf dem Modell von Wein et al. (WKC+07). Die Autoren nutzen für die Berechnungen der Schnittpunkte OptiX eine Ray-tracing Bibliothek für CUDA. Den Hauptteil des Papers bilden verschiedene Zeitmessungen. Es wird gezeigt, dass die Simulation auch mit relativ komplexen Mesh-Modellen echtzeitfähig ist. Auch mit mehreren Objekten steigt die benötigte Rechenzeit für eine Simulation nur sehr langsam an. Die Geschwindigkeitsmessungen für deformierte Objekte zeigt jedoch, dass hier nur relativ einfache Modelle in Echtzeit deformierbar sind. Dies liegt daran, dass OptiX Beschleunigungsstrukturen benötigt[2], die bei jeder Deformation neu berechnet werden müssen. Bereits bei 50 000 Dreiecken wird eine Berechnungszeit von mehr als einer Sekunde pro Bild angegeben. Eigene Zeitmessungen (siehe Kapitel 7) ergeben jedoch deutlich abweichende Werte. Dies liegt wahrscheinlich an der Wahl des Algorithmus, um die Beschleunigungsstrukturen zu erzeugen. OptiX stellt hierfür mehrere sogenannte „Acceleration builder“ zur Verfügung. In dem Paper wird diese Möglichkeit nicht erwähnt, daher wird wahrscheinlich der Standardalgorithmus verwendet, der für statische Mesh-Modelle optimiert ist.

[2] Ohne Beschleunigungsstrukturen liegt die benötigte Simulationszeit bereits bei sehr einfachen Mesh-Modellen bei mehreren Sekunden pro Bild.

4 Realisierung

Als geometrische Modelle für die Ultraschallsimulation wurden zwei verschiedene Modelle implementiert. Um einfache Schichtbilder mit fester Schallkopfeinstellung simulieren zu können, ohne ein aufwendiges 3D-Modell erstellen zu müssen, wurde eine 2D-Simulation entwickelt. Für Simulationen, bei denen der Schallkopf frei im Raum positionierbar sein soll, wurde ein 3D-Ray-tracing-Verfahren entwickelt. Das 3D-Ray-tracing-Verfahren kann auch in kompakter Form in dem Paper „Real-Time GPU-based Ultrasound Simulation Using Deformable Mesh Models“ nachgelesen werden (BBRH13).

4.1 2D-Ultraschall Simulation

Das Modell wurde hauptsächlich für die Simulation von intravaskulärem Ultraschall entwickelt, kann jedoch auch zur Simulation von anderen Ultraschallmodalitäten verwendet werden. Im Vergleich zum 3D-Ray-tracing-Verfahren besitzt es die folgenden Eigenschaften:

+ sehr leichte Modellerzeugung
+ geringer Rechenaufwand
− eingeschränkte Simulation von ultraschallspezifischen Artefakten
− eingeschränkte Lage des Schallkopfes

4.1.1 Modelle

Die 2D-Modelle bestehen aus mehreren Schichten. Eine Schicht repräsentiert eine bestimmte Z-Position des Schallkopfes. Jede Schicht besteht aus verschiedenen Objekten. Jedem Objekt wird ein Medium und eine Geometrie zugeordnet.

Das Medium beschreibt die zur Simulation benötigten Eigenschaften:

- den Absorptionskoeffizienten μ_a
- die Rauschverteilung, welche durch einen Rauschanteil μ_r, eine Standardabweichung μ_s und die Varianz σ definiert wird
- die Mediumübergangsvariablen γ_0, γ_1 und l_0, l_1

Die Geometrie beschreibt die äußere Form eines Objektes und wird als eine geschlossene Kontur modelliert. Eine geschlossene Kontur besteht aus drei oder mehr 2D-Koordinaten und der Startpunkt ist identisch mit dem Endpunkt.

$$P_n \in \mathbb{R}^2, \quad n \in \mathbb{N}, \quad n >= 3, \quad P_0 = P_n \tag{4.1}$$

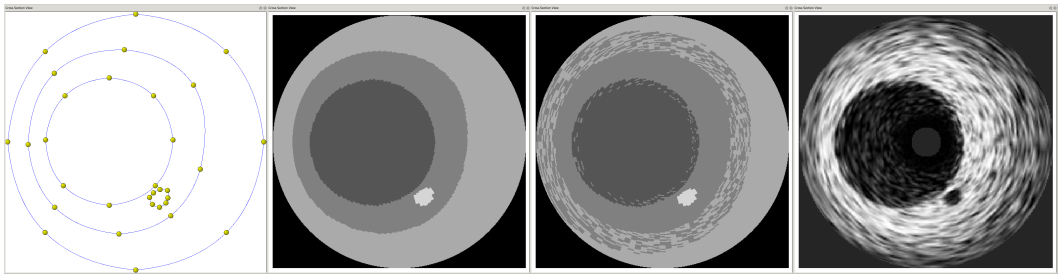
Jeder der Konturpunkte ist über eine Funktion mit seinem Nachfolger verbunden. Als Funktion können eine Gerade, eine Bézierkurve (parametrisch modellierte Kurve) oder ein Spline (stückweise zusammengesetzte Polynomfunktion) verwendet werden. Für mathematischen Definitionen und weiterführende Erklärungen zu Bézierkurven und Splines sei auf das Buch „Bézier and B-spline techniques“ (PBP02) verwiesen. In Abbildung 4.1a sind mehrere geschlossene Konturen dargestellt, die zusammen ein Gefäß modellieren.

Die Modelle werden mit einem selbst entwickelten Editor erzeugt, der in Abschnitt 6.1 genauer beschrieben wird.

4.1.2 Simulation

Die im Folgenden beschriebene 2D-Simulation von Ultraschall ist auf die Simulation von IVUS optimiert, kann aber auch für die Simulation anderer Ultraschallmodalitäten verwendet werden. Da die zu simulierenden Gewebearten einer IVUS Simulation alle eine annähernd gleiche Schallgeschwindigkeit haben, wird eine konstante Schallgeschwindigkeit angenommen. Für andere Ultraschallmodalitäten sollte diese Annahme überprüft werden. Haben alle durchschallten Medien die gleiche Schallgeschwindigkeit, dann gibt es keine Schallbrechung. Der Verlauf des Ultraschallimpulses kann daher als gerade angenommen werden. Ein aufwendiges Ray-tracing-Verfahren ist daher bei dieser vereinfachten 2D-Simulation nicht notwendig.

Um ein komplettes B-Mode Ultraschallbild zu berechnen, werden mehrere Strahlen entlang der Schallrichtung des virtuellen Schallkopfes in das Modell geschossen. Bei einem Linear-Array Schallkopf sind die Strahlen somit parallel und bei einem IVUS Schallkopf kreisförmig zueinander angeordnet. Ein Strahl wird mit Hilfe der Geradengleichung $f(x) = ax + c$ beschrieben. Der Startpunkt c ist durch die Position, an dem der Ultraschallpuls den Schallkopf verlässt, gegeben. Ebenso ist die Steigung a durch die Richtung des Ultraschallpulses festgelegt. Nun werden für jedes Objekt die Schnittpunkte mit der Geraden berechnet und in einer nach x sortierten Liste gespeichert. Ein Strahl, der ein Objekt kreuzt, hat i.d.R. eine gerade Anzahl von Schnittpunkten mit dem Objekt. Sollte ein Strahl eine ungerade Anzahl an Schnittpunkten mit einem Objekt haben, so ist ein Schnittpunkt eine Tangente und wird ignoriert. Es kann nun festgestellt werden, ob der Strahl in das Objekt eindringt oder es verlässt, indem die Anzahl der verbleibenden Schnittpunkte gezählt wird. Jeder Strahl wird äquidistant in Abhängigkeit von x abgetastet und das Medium an diesem Ort wird in einen Puffer



(a) (b) (c) (d)

Abbildung 4.1: Beispiel einer IVUS Simulation auf Basis eines 2D-Modells.

geschrieben. Das Ergebnis des Verfahrens wird in einen 2D-Puffer geschrieben, in dem die Abtastpunkte aller Strahlen stehen (Abbildung 4.1b). Zur Simulation eines weichen Gewebeüberganges wird an jedem Schnittpunkt eine Vermischung der beiden betroffenen Medien durchgeführt. Je höher der Übergangsfaktor γ eines Mediums ist, desto häufiger tritt das Medium innerhalb der Übergangslänge l auf. Abbildung 4.1c veranschaulicht das Ergebnis des Verfahrens. Im Anschluss daran wird der 2D-Puffer genutzt, um die Rückstreuung von Ultraschall in homogenen Medien mit Hilfe des Faltungsmodells von Bertrand und Meunier (MB95) zu berechnen. Das Faltungsmodell wird in Abschnitt 4.2.2 in einer modifizierten 3D-Variante genauer beschrieben. Nach Berechnung der Rückstreuung lässt sich das Ultraschallbild darstellen, indem man die Rückstreuung aller Abtastpunkte als Grauwerte interpretiert und nebeneinander anordnet (Abbildung 4.1d). Analog zu echten Ultraschallsystemen werden vorher noch einige Bildnachbearbeitungsschritte (siehe Abschnitt 4.3) ausgeführt, um die Bildqualität zu verbessern.

4.2 3D-Ultraschall Simulation

Um ein realistisches Ultraschallbild mit typischen Ultraschallartefakten zu simulieren wird ein 3D-Modell benötigt. Im Folgenden werden diese Modelle und anschließend die Simulation auf Basis der Modelle beschrieben.

4.2.1 Modelle

Ein 3D-Modell besteht aus mehreren Objekten, denen ultraschallspezifische Eigenschaften zugewiesen werden. Das Modell beschreibt die Geometrie des zu schallenden Objektes. Innerhalb des 3D-Raumes muss jedem Punkt im Raum, $X \rightarrow (x, y, z)\ \ x, y, z \in \mathbb{R}$, genau ein Objekt zuordenbar sein. Es existieren verschiedene Modelle, um solche geometrischen 3D-Objekte zu beschreiben. Die 3D-Objekte können analytisch, über Volumenelemente oder über Oberflächenmodelle beschrieben werden. Da die analytische Beschreibung komplexer Modelle

sehr aufwendig ist, werden für die meisten 3D-Modelle Volumenelemente oder Oberflächenmodelle verwendet. Volumenelemente verwenden i.d.R. mehr Punkte als Oberflächenmodelle, um ein Objekt zu beschreiben. Dadurch benötigen Volumenmodelle mehr Arbeitsspeicher und dadurch bedingt meistens auch mehr Rechenkapazität. Oberflächenmodelle haben darüber hinaus den Vorteil eine leichte Nachbearbeitung der Modelle zu ermöglichen. Der Nachteil der Oberflächenmodelle, zumindest im medizinischen Bereich, ist die kompliziertere Erstellung der Modelle. Während realistische Volumenmodelle mit Hilfe von bildgebenden Verfahren wie CT, MRT oder PET erzeugt werden können, gestaltet es sich bei Oberflächenmodellen schwieriger, realistische Modelle zu erzeugen. In Kapitel 6.2 werden die verschiedenen Möglichkeiten beschrieben, geeignete Oberflächenmodelle zu generieren. Aufgrund der vielen Vorteile von Oberflächenmodellen gegenüber Volumenmodellen werden heutzutage im Bereich der Computergrafik fast ausschließlich Oberflächenmodelle verwendet.

Oberflächenmodelle können über verschiedene Verfahren beschrieben werden. Polygonmodelle, auch Gitternetze oder Mesh-Modelle genannt, sind eine Darstellung der Objektoberfläche durch eine endliche Menge an polygonalen, planaren, verbundenen Flächen. Freiformkurven bzw. -flächen (auch parametrische Kurven und Flächen genannt) verwenden stückweise verbundene Kurven bzw. Flächen, um eine Oberfläche zu beschreiben. Die parametrische Modellierung wird häufig beim Erstellen eines Modells verwendet. Für die Visualisierung oder Kollisionserkennung werden sie jedoch in Polygonmodelle umgewandelt. Ein Polygon besteht aus drei oder mehr Punkten, die jeweils mit ihrem Nachfolger über eine Gerade verbunden sind. Der Start- und Endpunkt ist auch über eine Gerade miteinander verbunden, wodurch sich eine geschlossene Fläche ergibt. Jedes Polygon lässt sich durch Triangulationsalgorithmen in eine Teilmenge von Dreiecken einteilen, welche zusammen die komplette Fläche des Polygons abdecken. Somit lässt sich jedes beliebige 3D-Objekt nur mit Hilfe von Dreiecken beschreiben. Dies vereinfacht die Visualisierung und Kollisionserkennung, da alle Operationen, die benötigt werden, nur für Dreiecke optimiert sein müssen.

4.2.2 Physik des Ray-tracings

Da die exakte Simulation einer Schallwelle, wie in Abschnitt 3.2.2 beschrieben, rechnerisch für eine Echtzeitsimulation zu aufwendig ist, wird eine Approximation verwendet. Wie in Kapitel 2.2 beschrieben, kann, vom geometrischen Blickwinkel gesehen, die Schallwelle als Lichtstrahl behandelt werden. Eine Schallwelle wird dann durch mehrere dicht gebündelte Strahlen approximiert, um die Ausdehnung der Schallwelle zu berücksichtigen. Der Verlauf eines Strahles kann mit Hilfe der optischen Brechungsgesetze berechnet werden. Ausgehend vom Schallkopf wird der Strahl durch das 3D-Modell geschossen. Während der Strahl durch das Modell läuft, wird er aufgrund von interner Reibung, Relaxation und Streuung

absorbiert. Dies führt zu einer Abschwächung der Intensität, welche mit dem Beer-Lambert-Gesetz 2.8 berechnet wird.

Wenn ein Strahl auf die Grenzschicht zwischen zwei Objekten trifft, kann er gebrochen und reflektiert werden. Trifft ein Strahl unter dem Winkel θ_1 auf die Grenzschicht, dann wird er um den Winkel θ_2 gebrochen und um den Winkel θ_3 reflektiert (siehe Abbildung 4.2a). Durch Verwendung des Snelliusschen Brechungsgesetzes 2.5 ergeben sich die Gleichungen,

$$\cos\theta_1 = N\,(-V_i) \tag{4.2}$$

$$\cos\theta_2 = \sqrt{1 - \left(\frac{Z_1}{Z_2}\right)^2 (1 - \cos\theta_1^2)} \tag{4.3}$$

$$V_r = V_i + (2\cos\theta_1)\,N \tag{4.4}$$

$$V_t = \frac{Z_1}{Z_2} V_i + \left(\frac{Z_1}{Z_2}\cos\theta_1 - \cos\theta_2\right) N \tag{4.5}$$

mit denen die Richtungen des gebrochenen Strahles V_t und des reflektieren Strahles V_r berechnet werden können. Die benötigten Variablen zur Berechnung sind die aktuelle normalisierte Richtung v_i des Strahles, die normalisierte Normale N der Grenzschicht und die Schallimpedanzen Z_1 und Z_2 der beiden Medien.

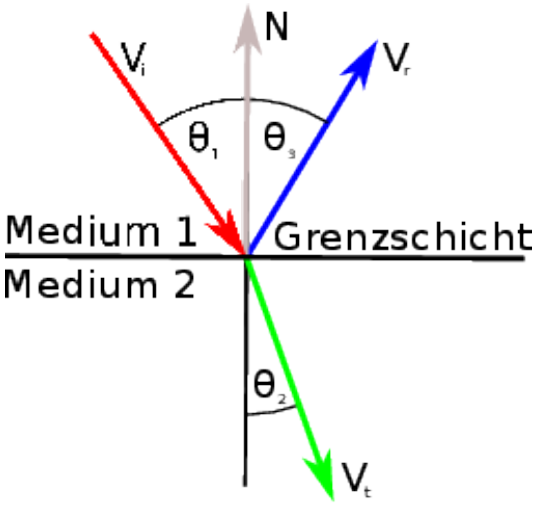
Die neuen Intensitäten der beiden Strahlen werden mit Hilfe von Fresnels Gleichungen 2.7 berechnet. Ersetzt man den Brechungs- und Reflexionskoeffizienten durch die Intensität so ergeben sich folgende Formeln:

$$I_r = I_i \left(\frac{Z_1\cos\theta_1 - Z_2\cos\theta_2}{Z_1\cos\theta_1 + Z_2\cos\theta_2}\right)^2 \tag{4.6}$$

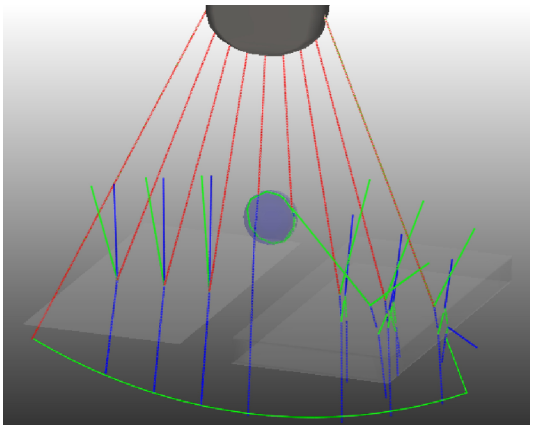
$$I_t = I_i \frac{4Z_1\cos\theta_2 Z_2\cos\theta_1}{Z_1\cos\theta_1 - Z_2\cos\theta_2} \tag{4.7}$$

Um nicht benötigte Berechnungen zu minimieren, wird die Berechnung des Strahles gestoppt, sobald eine maximale Länge oder minimale Intensität erreicht ist.

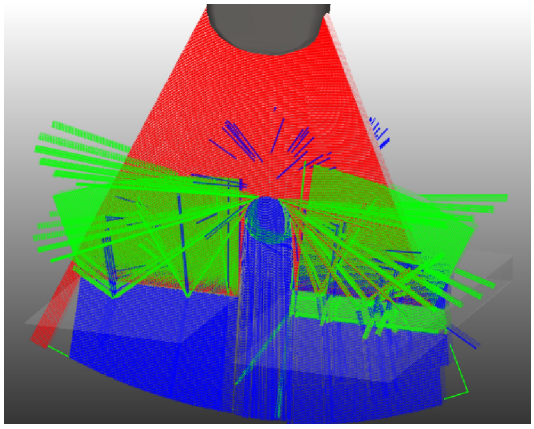
Der beschriebene Algorithmus benötigt für seine Berechnungen immer das letzte sowie das aktuelle Material. Ein Material wird immer dann aktiv, wenn die Außenseite des zugehörigen Polygonnetzes getroffen wird. Es wird immer dann inaktiv, wenn die Innenseite getroffen wird. Alle aktiven Materialien werden in einem Stack gespeichert beziehungsweise wieder aus diesem gelöscht. Die letzten beiden Elemente sind dann immer die beiden zu verwendenden Materialien. Dieser Algorithmus funktioniert allerdings nicht bei Mesh-Modellen, die sich gegenseitig überschneiden. Bei diesen kann nicht eindeutig bestimmt werden, welches Material aktiv sein soll. Eine gegenseitige Überschneidung von Gitternetzen sollte zwar bei der Mesherstellung vermieden werden, jedoch lässt sich dies nicht immer verhindern. Blutgefäße oder auch das Lymphsystem verlaufen durch mehrere



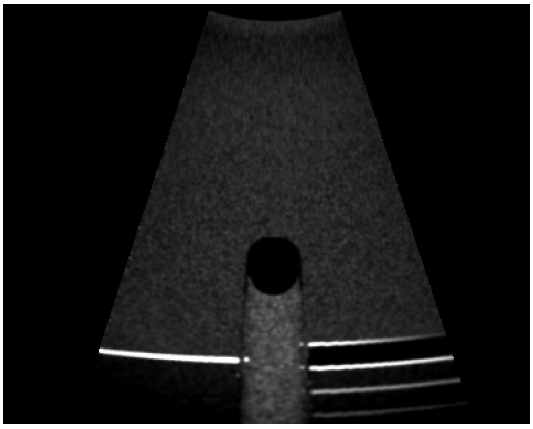
(a) Brechung und Reflexion eines Strahles an einer Grenzschicht.



(b) Vereinfachtes Ray-tracing Szenarium mit neun Strahlen. Innerhalb der Kugel ist eine interne Reflexion sichtbar.



(c) Komplettes Ray-tracing Szenarium mit 5376 Strahlen.



(d) Simuliertes Ultraschallbild basierend auf den Szenarien in Abbildung (b)-(c)

Abbildung 4.2: Die Quader und die Kugel sind in den Bildern transparent dargestellt. Alle Objekte haben eine hohe Impedanz verglichen mit ihrer Umgebung. Alle Strahlen, die direkt vom Schallkopf abgehen, sind in rot, gebrochene Strahlen in blau und reflektierte Strahlen in grün dargestellt.

Organe. An den Grenzschichten gibt es daher zwangsläufig Überschneidungen oder die Mesherzeugung und das Mesh selbst werden um einige Größenordnungen komplexer. Auch bei der Simulation von Instrumenten kann nicht gewährleistet werden, dass keine Überschneidungen auftreten. Daher wird für jeden Materialtyp eine Ordnungszahl eingeführt. Die Materialien auf dem Stack mit den beiden höchsten Ordnungszahlen werden dann für den Algorithmus verwendet.

Eine Alternative zu dem eben beschriebenen Algorithmus zur Bestimmung der aktiven Strahlen wäre es, jedem Dreieck unterschiedliche Materialtypen für die Vorder- und Rückseite zuzuordnen. Der zusätzlich benötigte Speicher beträgt zwei Bytes pro Dreieck und wäre vertretbar. Es müsste ein Algorithmus geschrieben werden, der die Materialtypen für jedes Dreieck automatisch bestimmt. Des Weiteren müssten alle Schnittpunktprogramme 5.3.2 modifiziert werden. Da der Mehraufwand in keiner Relation zum erwarteten Geschwindigkeitsvorteil steht, wurde diese Alternative nicht implementiert.

Ein Ray-tracing ist sehr rechenaufwendig und muss parallelisiert werden, damit die Bilder in Echtzeit simuliert werden können. Da der Simulator auf einem Standard-Rechner laufen soll, entfällt eine Cluster-Lösung. Eine Parallelisierung auf der CPU würde maximal eine Beschleunigung um den Faktor acht bringen. Allerdings nutzen sowohl die Berechnung der Deformation als auch die haptische Eingabe die CPU, wodurch dieser Faktor deutlich geringer ausfallen würde. Da für alle Strahlen die gleichen Berechnungen gemacht werden müssen, kann dieser Teil optimal parallelisiert und komplett auf die Grafikkarte (GPU) ausgelagert werden. Das komplette Ray-tracing wurde mit Hilfe des in Kapitel 5.3 beschriebenen Frameworks OptiX auf die GPU portiert.

Echoerzeugung

Ein Ultraschallecho entsteht aufgrund der Reflexion an Makrostrukturen und der Rückstreuung des Signales an Mikrostrukturen (siehe Kapitel 2.2). Die Reflexion entsteht nur, wenn der Schallwiderstand der beiden Objekte unterschiedlich ist und ist spiegelnd, diffus oder in den meisten Fällen eine Kombination aus beiden. Beide Terme werden, um eine schnelle Berechnung zu gewährleisten, approximiert. Während die spiegelnde Reflexion mit Gleichung 2.7 gut approximiert werden könnte, würde die diffuse Reflexion vernachlässigt werden. Um sowohl die spiegelnde als auch die diffuse Reflexion zu berücksichtigen, wird eine modifizierte Version von Lamberts' Kosinusgesetz $\frac{I_r}{I_i} = \cos\theta_1$ verwendet,

$$\frac{I_r}{I_i} = \cos\theta_1{}^n \tag{4.8}$$

wobei I_r die Intensität des reflektierten Signales ist, I_i die Intensität des Strahles am Schnittpunkt und n das Verhältnis von spiegelndem zu diffusem Echo angibt. Ähnlich zu der Ultraschallsimulation von Wein et al. (WKC+07) werden auch in

dieser Simulation die relativen Schallimpedanzunterschiede verwendet. Der Faktor n wird aber behalten, um das Verhältnis von spiegelnder Reflexion (z. B. Knochen) zu diffuser Reflexion beeinflussen zu können.

$$I_r \approx \left| \cos\theta_1{}^n * \left(\frac{Z_2 - Z_1}{Z_1 + Z_2} \right)^2 * (I_i)^\alpha \right| \tag{4.9}$$

Zusätzlich wird die Variable α genutzt, um kleine Reflexionen verstärken zu können.

Rückstreuung entsteht an jeder Stelle des Ultraschallstrahles und wird über eine modifizierte Version des Faltungsmodells von Bertrand und Meunier (MB95) berechnet. Anstelle einer 2D-Version, wird eine 3D-Variante genutzt, um die elevationale Auflösung (siehe Kap. 2.3.3) des zu simulierenden Ultraschallsystems zu berücksichtigen (MV09). Die Faltung wurde modifiziert, um die Intensität I der Schallwelle an der aktuellen Abtastposition zu berücksichtigen,

$$E(X) = P(X)\, I(X) \otimes \widetilde{Z}(X) \tag{4.10}$$

wobei E das am Schallkopf empfangende Echo ist, P die Pulsfunktion, und $\widetilde{Z}$ die Impedanzfunktion. Der Faltungsoperator wird mit $\otimes$ bezeichnet, und die aktuelle räumliche Position des Abtastpunktes mit $X \to (x, y, z)\ \ x, y, z \in \mathbb{R}$. Die Pulsfunktion, auch bekannt als Punktantwort, Punktbildfunktion (engl. ***p**oint **s**pread **f**unction*, PSF, Abb. 4.3) wird mit einer dreidimensionalen Gaußfunktion, moduliert mit einer Kosinusfunktion, approximiert.

$$P(X) = \exp\left(-\frac{1}{2} \left(\frac{x^2}{\sigma_x^2} + \frac{y^2}{\sigma_y^2} + \frac{z^2}{\sigma_z^2} \right) \right) g(x) \tag{4.11}$$

$$g(x) = \cos(2\pi f x) \tag{4.12}$$

Die Impedanzfunktion $\widetilde{Z}$ definiert die Anzahl, Verteilung und Beschaffenheit der Streukörper in einem Objekt (FBC99). Sie wird über zwei Wahrscheinlichkeitsdichtefunktionen beschrieben, eine für die Anzahl und die andere für die Verteilung und Beschaffenheit der Streukörper. Da eine komplette 3D-Faltung für eine Echtzeitanwendung zu rechenintensiv ist, wird die Faltung in drei 1D-Faltungen aufgeteilt, je eine pro Dimension (axial, lateral und elevational). Dieser Ansatz ist ähnlich zu der von Gao et al. (GCC+09) beschriebenen COLE-Methode, wurde aber um eine Dimension erweitert. Theoretisch wäre es zwar möglich eine noch genauere PSF zu nutzen, aber diese könnte nicht mehr in drei einzelne Faltungen aufgeteilt werden. Dadurch würde der Rechenaufwand so stark steigen, dass eine Berechnung in Echtzeit nicht mehr möglich wäre. Der größte Unterschied zu einer exakteren PSF wäre im Nahfeld, das für den diagnostischen Ultraschall keine große Rolle spielt. Für das Fernfeld ist die Approximation der PSF als eine durch einen Kosinus modulierte 3D-Gaußfunktion ausreichend (MB95).

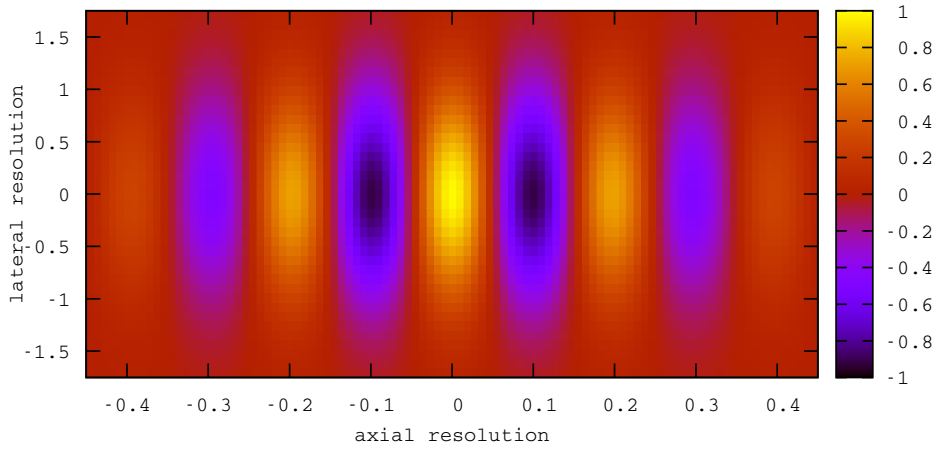


Abbildung 4.3: Typische Punktbildfunktion eines 2,5 MHz Schallkopfes. Zur Übersichtlichkeit werden nur die axiale und laterale Ausrichtung visualisiert.

Rauschtextur

Wird der Schallkopf langsam bewegt so verändert sich auch die Speckle-Struktur aufgrund der räumlichen Ausdehnung der Impulsfunktion, nur langsam. Um diesen Effekt zu simulieren, müssen alle Streukörper gespeichert werden. Allerdings wäre es sehr speicheraufwendig, alle Streukörper der kompletten Simulation zu speichern. Sogar eine kleine Simulation, wie das BAT-Phantom (Kapitel 7.2.1), benötigt einen Streukörper von mindestens $1024 \times 1024 \times 1280$ Streupunkten. Der benötigte Speicher würde nur auf eine Hochleistungsgrafikkarte passen und größere Simulationen wären nicht möglich. Nutzt man die Eigenschaft aus, dass ein Punkt im Raum zwar ein eindeutiges Streuverhalten haben muss, aber nicht jedes Streuverhalten eindeutig einem Punkt zuzuordnen sein muss, so kann der zu speichernde Streukörper relativ klein gehalten werden. Zum Speichern des Streukörpers wird eine 3D-Textur verwendet (s. Abb. 4.4), welche sich wiederholt sobald der Streukörper verlassen wird. Es gilt also $T(X) = T(sX)$ wobei T die 3D-Textur bezeichnet und s die Größe der Textur. Die optimale Größe von s wurde empirisch ermittelt, indem alle realistischen Texturgrößen untersucht wurden. Da s aufgrund von Hardwarelimitationen eine Zweierpotenz sein muss, gilt $s = 2^p, p \in \mathbb{N}$. Zu kleine Potenzen können ausgeschlossen werden, da hierbei sich wiederholende Muster in dem simulierten Bild zu erkennen sind. Zu große Potenzen können ebenfalls ausgeschlossen werden, da eine 3D-Textur, die größer als der zu speichernde Streukörper ist, keinen Vorteil mehr bringt. Somit bleiben nur

fünf mögliche Texturgrößen ($s \in 32, 64, 128, 256, 512$). Aus diesen Texturgrößen muss die kleinste gefunden werden, bei der keine Wiederholungen innerhalb des simulierten Bildes zu sehen ist. Die meisten Wiederholungen gibt es, wenn die Strahlen senkrecht zu einer der drei Koordinatenachsen verlaufen, da hier alle Streupunkte mehrmals verwendet werden. Doch selbst in einem solchen Fall, der selten in der Realität auftritt, konnten mehrere Bildverarbeitungsspezialisten die Texturwiederholungen nicht visuell erkennen, sobald eine höhere Texturgröße als 128×128×128 verwendet wurde.

Die Rauschtextur T besteht aus zwei normalisierten gaußverteilten Zufallszahlen. Diese werden zusammen mit den gewebeabhängigen Parametern ν, μ und σ der Wahrscheinlichkeitsdichtefunktion genutzt, um die Impedanz $\widetilde{Z}(X)$ des Objektes am Punkt X zu berechnen.

$$S(X) = T\,(X) * \sigma\,(X) + \mu\,(X) \tag{4.13}$$

$$\widetilde{Z}\,(X) = \begin{cases} S(X) & T(X,1) \leq \nu(X) \\ 0 & \text{otherwise} \end{cases} \tag{4.14}$$

Die Funktion $T(X)$ nutzt noch eine Skalierungsvariable, bevor auf die Textur zugegriffen wird. Mit Hilfe dieser Variablen kann das Aussehen, insbesondere die Größe, der Speckles beeinflusst werden. Die Größe ist linear abhängig von der gewählten Frequenz des Schallkopfes, da mit zunehmender Frequenz die Auflösung des Ultraschallbildes steigt. Man kann das Aussehen der Speckle-Struktur noch mit einer kleinen Modifikation weiter beeinflussen. Indem mehrere Texturen mit unterschiedlichen Skalierungsvariablen verwendet werden, kann man unterschiedlich große Strukturen simulieren, obwohl für diese keine Geometrie vorliegt. So kann man zum Beispiel simulieren, dass die Leber in bestimmten Bereichen dichter ist als in anderen. Der Nachteil dieses Verfahrens ist, dass die entstandenen Strukturen komplett zufällig sind. Im Kapitel 5.4 werden weitere Details zur 3D-Textur erläutert.

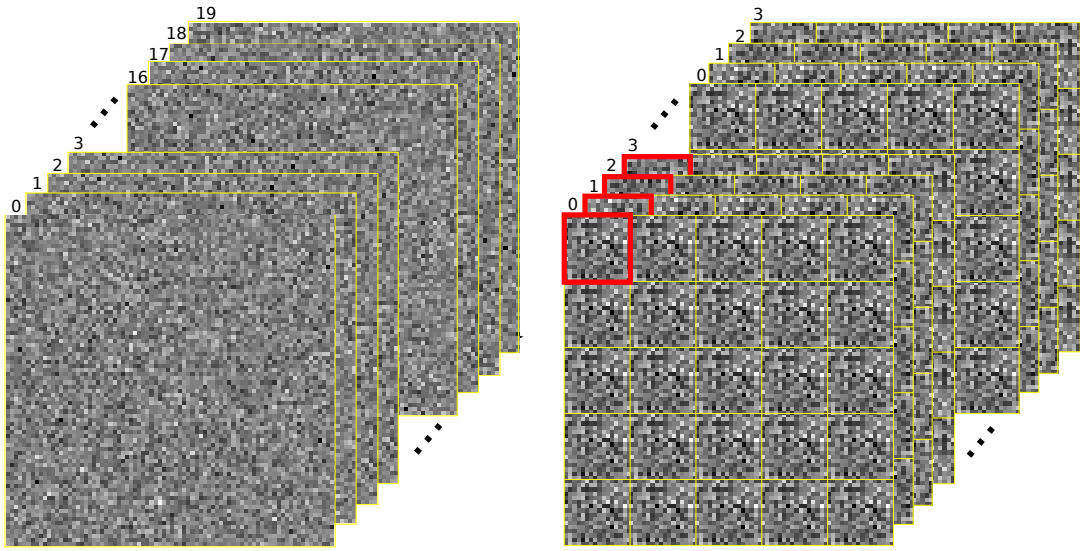


Abbildung 4.4: Links: Kompletter Streukörper bei dem alle Elemente gespeichert werden. Rechts: Vereinfachter Streukörper. Nur die Elemente in den roten Vierecken werden gespeichert.

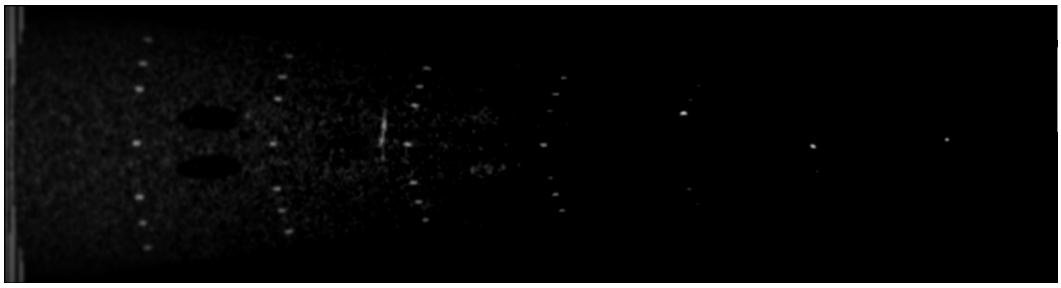
4.3 Nachbearbeitung

Nach der Echoerzeugung lassen sich alle Abtastpunkte als ein 2D-Bild darstellen (Abbildung 4.5a). Analog zu echten Ultraschallsystemen wird das Bild auch in der Simulation nachbearbeitet, um die optische Qualität zu verbessern.

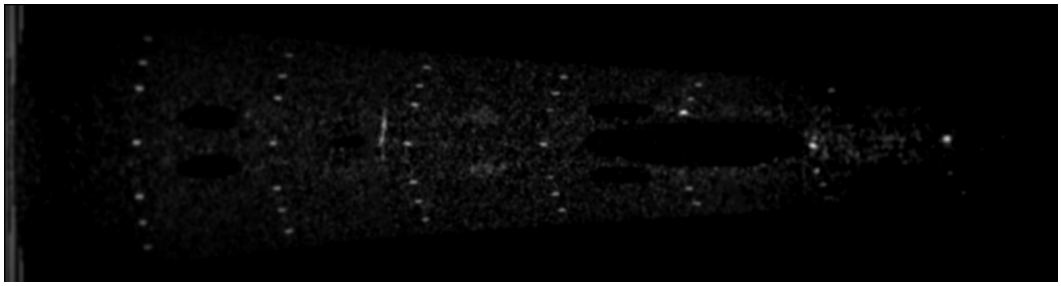
In echten Ultraschallsystemen wird eine ***t****ime* ***g****ain* ***c****ompensation* (TGC) ausgeführt, um die Dämpfung des Gewebes auszugleichen (Abbildung 4.5b). Diese wird automatisch ausgeführt, kann jedoch auch am Gerät direkt eingestellt werden. Um dem Anwender der Simulation die gleichen Bearbeitungsmöglichkeiten der TGC zu ermöglichen, kann die Intensität aller Abtastpunkte in Abhängigkeit zur Zeit verändert werden. Die Parameter der TGC werden über einen Spline eingestellt, dessen Kontrollpunkte beliebig verändert werden können.

Wie in Kapitel 2.3.5 beschrieben, existieren verschiedene Arten von Schallköpfen. In dem Simulator wurden ein Linear-Array Schallkopf, ein Konvex-Array Schallkopf, ein IVUS Schallkopf und ein transrektaler Ultraschallkopf implementiert. Die Simulationsergebnisse liegen alle linear im Speicher. Um die empfangenen Schallimpulse analog zur Ausbreitungsrichtung der Schallimpulse darzustellen, muss eine Koordinatentransformation durchgeführt werden. Das Ergebnis dieser Koordinatentransformation ist in Abbildung 4.5c zu sehen und wird in Kapitel 5.5 genauer beschrieben.

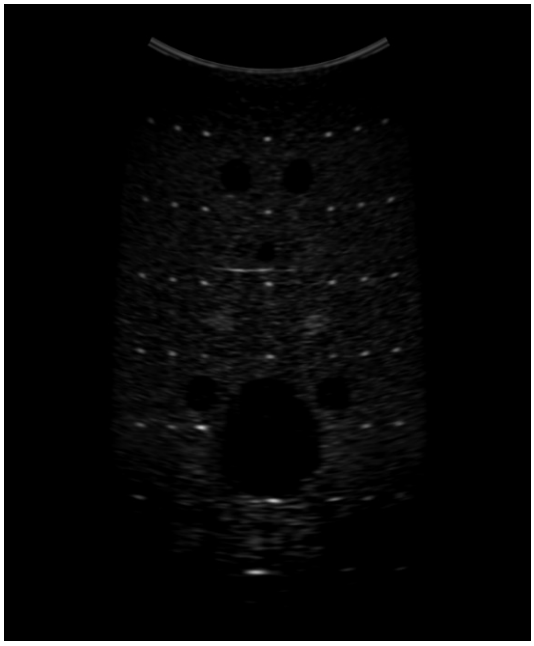
Das menschliche Auge kann große Farbunterschiede besser wahrnehmen als kleine Farbunterschiede. Um kleine, definierte Farbbereiche besonders hervorzuheben, können diese mit Hilfe einer Gradationskurve verstärkt werden. Die



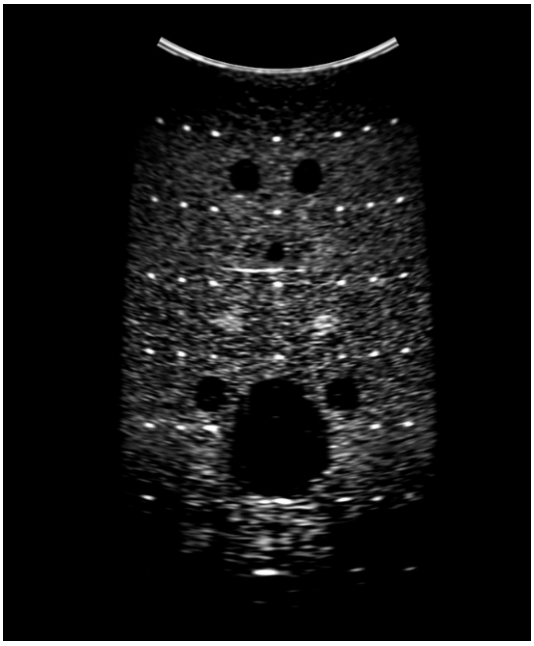
(a) Das Originalbild nach der Echoerzeugung



(b) Mit *time gain compensation*



(c) Koordinatentransformation von Polar- zu kartesischen Koordinaten



(d) Nach Anwendung einer Gradationskurve

Abbildung 4.5: Bildnachbearbeitungsschritte angewendet auf das simulierte Bild in der Reihenfolge (a)-(d).

Gradationskurve bestimmt die Darstellung der Grauwerte auf dem Bildschirm. Jedem Grauwert $G(x)$ wird somit in Abhängigkeit einer Funktion G_g ein neuer Wert zugewiesen:

$$G_{neu}(x) = G_g\left(G\left(x\right)\right) \tag{4.15}$$

Auch hier wird ein Spline verwendet, um den Kurvenverlauf leicht und benutzerfreundlich anpassen zu können. Alle beschriebenen Schritte sind in CUDA implementiert und werden auf der GPU ausgeführt.

5 Implementierung

Um die Simulation in Echtzeit berechnen zu können, muss sie sehr effizient programmiert sein und auf einem schnellen Rechner laufen. Da sie auf einem Standard-PC laufen soll, entfällt die Möglichkeit, sie auf einem Computercluster laufen zu lassen. Eine weitere Möglichkeit ist die Grafikkarten-Programmierung mit einer GPGPU-Programmiersprache (engl. ***G**eneral **P**urpose Computation On **G**raphics **P**rocessing **U**nits*). Hierbei wird die GPU zweckentfremdet und für allgemeine Berechnungen nicht grafiklastiger Berechnungen verwendet. Die GPU führt eine Aufgabe i.d.R. langsamer als eine CPU aus, jedoch kann eine GPU sehr viele Threads parallel verarbeiten. Während aktuelle Standardrechner bis zu acht Threads gleichzeitig bearbeiten können, besitzen aktuelle Grafikkarten bis zu 3072 Recheneinheiten. Des Weiteren können sehr viele Threads im Speicher gehalten werden, wodurch Speicherladezeiten überbrückt werden können, indem zwischen aktiven und inaktiven Threads gewechselt wird. Die Threads führen die gleiche Anweisung mit unterschiedlichen Daten aus. Dieses Verfahren ist in der Informatik als SIMD (engl. ***S**ingle **I**nstruction **M**ultiple **D**ata*) bekannt [1].

In den Anfangszeiten der GPGPU war die Programmierung schwierig und nur eingeschränkt möglich. Bestehende Shadingsprachen wie Cg, HLSL und GLSL wurden zweckentfremdet, um bestimmte, nicht grafikorientierte Algorithmen auf der GPU zu berechnen. Mittlerweile hat die Grafikkarten-Industrie die weiteren Anwendungsmöglichkeiten erkannt und die Grafikkarten werden auch in Hinblick auf die GPGPU weiterentwickelt. Es existieren drei Standards, welche die Programmierung der Grafikchips deutlich vereinfachen: OpenCL, CUDA C und DirectCompute (C++ AMP). OpenCL ist ein offener Standard, um CPUs, GPUs und andere Geräte von verschiedenen Herstellern zu programmieren. CUDA (engl. ***C**ompute **U**nified **D**evice **A**rchitecture*) ist eine von Nvidia entwickelte Architektur, die mit verschiedenen Sprachen programmiert werden kann. Die Sprache mit der CUDA am flexibelsten programmiert werden kann ist CUDA C und wird in Abschnitt 5.2 genauer beschrieben. DirectCompute ist eine von Microsoft entwickelte Erweiterung zu DirectX, die auf HLSL basiert. Da die Verwendung von DirectCompute[2] auf Windowssysteme beschränkt ist und es im Vergleich zu den beiden Alternative schwieriger zu nutzen und weniger verbreitet ist, sind

[1] Nvidia hat einen neuen Begriff namens SIMT für die GPU-Programmierung eingeführt (engl. ***S**ingle **I**nstrumention **M**ultiple **T**hreads*). Allerdings ist dieser Begriff, der für die Ausführung von mehreren Prozessen mit nur einem Befehl steht, in der Fachwelt noch nicht akzeptiert

[2] (Mitte 2012 hat Microsoft die Programmbibliothek C++ AMP vorgestellt, welche im Laufe der Zeit DirectCompute wahrscheinlich ersetzen wird.)

OpenCL und CUDA die besseren Alternativen.

Da CUDA vor OpenCL entwickelt wurde, ist es etwas verbreiteter und es existieren mehr Programmbibliotheken, die genutzt werden können. Zum Zeitpunkt der Entwicklung des Simulators war CUDA etwas schneller im Vergleich zu OpenCL (KDH10). Aus diesem Grund wurde dieser Simulator mit Hilfe von CUDA entwickelt. Mittlerweile hat OpenCL in Bezug zur Geschwindigkeit aufgeholt und die beiden Sprachen sind in etwa gleich schnell (FVS11).

5.1 Programmdesign

Um den Programmcode so verständlich und flexibel wie möglich zu implementieren, wurden die folgenden Möglichkeiten angewendet. Die Programmierung erfolgt unter Beachtung von objektorientierten Designtechniken. Die Klassen wurden in verschiedene Software-Bibliotheken unterteilt um eine bessere Übersicht über den Programmcode zu bekommen und die Wiederverwendbarkeit des Codes zu ermöglichen. Diese Software-Bibliotheken werden im nächsten Abschnitt beschrieben. Generische Programmierung wurde bei allgemeinen Klassen und Funktionen angewendet. Es wurde bewusst auf übermäßigen Gebrauch dieser Technik verzichtet, da generischer Code schlechter wartbar ist und längere Kompilierzeiten hat. Des Weiteren wurden die für C++ üblichen Entwurfsmuster (engl. *design patterns*) verwendet. Diese Entwurfsmuster werden sehr gut in dem Buch „Design Patterns“ von der „*Gang of Four*“ beschrieben (GHJV95). Die Patterns wie Skeleton, Command oder MFC sind als solche in dem Programmcode gut erkennbar und erleichtern die Verwendung und das Verständnis des Programmcodes. Die Software-Bibliothek Qt (Qt) wurde für mehrere Programmfunktionalitäten verwendet, da sie performant, objektorientiert und kompatibel zu Windows, Mac OS und Linux ist. Die einzelnen Programmteile sind:

- Die graphische Benutzeroberfläche (kurz GUI), welche durch eine ereignisgesteuerte Programmierung mit Hilfe des Signal und Slot Konzeptes programmiert wurde.
- Die Thread-Verwaltung welche in Abschnitt 5.6.1 genauer beschrieben wird.
- Der XML-Parser für das Einlesen eines Szenariums 6.4.
- Allgemeine Funktionen wie Container, Smartpointer und die OpenGL-Shader Verwaltung.

5.1.1 Eigene Software-Bibliotheken

In Abbildung 5.1 ist der Programmaufbau der beiden Simulatoren schematisch dargestellt. Der Programmcode für die Simulatoren ist in mehrere gekapselte Software-Bibliotheken aufgeteilt.

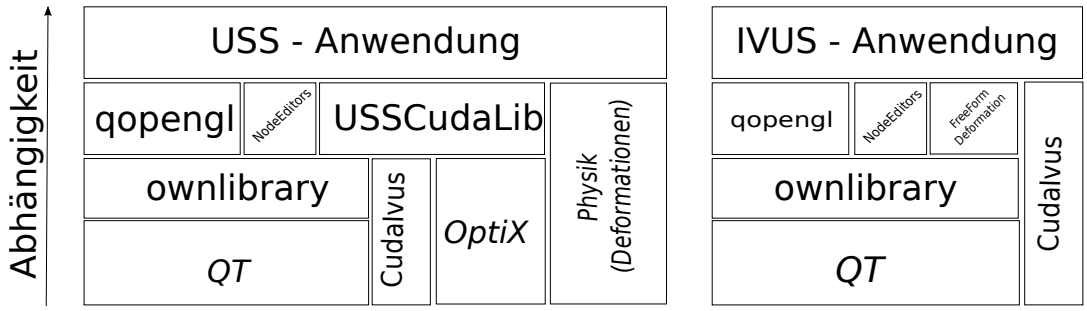


Abbildung 5.1: Schematischer Aufbau der beiden Ultraschallsimulatoren.

Die Basisbibliothek, welche von allen anderen Bibliotheken genutzt wird, ist die Bibliothek *ownlibrary*. Hier sind alle Hilfsklassen und -funktionen enthalten, die für alle anderen Bibliotheken, aber auch für andere in dieser Dissertation nicht beschriebenen Programme programmiert wurden. Beispiele sind Funktionen zum Laden und Speichern, Zufallsgeneratoren, Zeitmessungsfunktionen, allgemeine neue Qt-Widgets und mathematische Funktionen.

Die Bibliothek *qopengl* stellt Funktionen und Klassen zur Verfügung, um 3D-Objekte zu laden, speichern, modifizieren und darzustellen. Des Weiteren enthält sie alle Programmfunktionen, die auf OpenGL-Funktionen zurückgreifen.

Die Bibliothek *NodeEditors* enthält Klassen, um interaktiv eine Funktion zwischen mehreren frei modifizierbaren Stützstellen zu interpolieren. Neben der linearen Interpolation können Polygone oder verschiedene Splinetypen verwendet werden.

Die Bibliothek *CudaIvus* wird für alle Berechnungen verwendet, die direkt auf CUDA C basieren. Dies sind vor allem die Faltung und die Nachbearbeitungsschritte der Simulation.

Die Bibliothek *USSCudaLib* ist die Hauptbibliothek zur Simulation der 3D-Ultraschallbilder. Der komplette Ray-tracing-Prozess ist in dieser Bibliothek implementiert. Daher befindet sich hier aller Code, der die Ray-tracing-Bibliothek OptiX nutzt.

Die Anwendung *USS* ist der Kleber des 3D-Simulators, der alle verwendeten Bibliotheken miteinander verbindet. Außerdem ist sie verantwortlich für die grafische Oberfläche. Alle Nutzereingaben werden hier verarbeitet und an die jeweiligen Bibliotheken weitergegeben.

Die Anwendung *IVUS* ist der Kleber des 2D-Simulators, der alle verwendeten Bibliotheken miteinander verbindet. Außerdem ist sie verantwortlich für die grafische Oberfläche. Alle Nutzereingaben werden hier verarbeitet und an die jeweiligen Bibliotheken weitergegeben.

Die Bibliotheken bestehen aus über 40 000 Zeilen C++-Code und über 2500 Zeilen Cuda-Code. Leer- und Kommentarzeilen wurden nicht mit berücksichtigt.

5.2 CUDA C

Im Folgenden wird die Funktionsweise der Grafikkartenprogrammierung mit Hilfe von Nvidias Programmiersprache CUDA C kurz skizziert. Eine ausführliche Beschreibung bietet der kostenlose *NVIDIA CUDA C Programming Guide* (Cor11).

CUDA C ist eine von Nvidia entwickelte Erweiterung der Programmiersprache C zur Programmierung von CUDA-fähigen Grafikkarten [3]. Die Sprache kommt mit einer kleinen Erweiterung der Sprache C, aus wodurch sie bei vorhandenen C-Kenntnissen sehr schnell erlernt werden kann. Über die Erweiterung kann der Programmierer sogenannte Kernels definieren, die parallel von CUDA Threads ausgeführt werden. Die Anzahl der Threads kann der Programmierer im Rahmen der leichteren Verwendbarkeit ein-, zwei- oder dreidimensional angeben. Eine unterschiedliche Abarbeitung der Threads kann durch einen dreidimensionalen Indexvektor erreicht werden, der für jeden Thread einen eindeutigen Index zuordnet. Ein Kernel arbeitet besonders effizient, wenn benachbarte Threads auf benachbarte Speicherstellen zugreifen, also ein Thread mit dem Index (0,0,0) auf die Speicherstelle X und der benachbarte Thread mit dem Index (1,0,0) auf die Speicherstelle $X + 1$. Darüber hinaus gibt es sehr viele weitere Kernel-Optimierungstechniken, die ausführlich im *CUDA C Best Practices Guide* (NVI11) beschrieben werden.

CUDA ist die derzeit am meisten genutzte Programmiersprache für GPGPU-Programmierung und unter Linux, Mac OS und Windows lauffähig. Ein Nachteil von CUDA C im Vergleich zu anderen GPU-Programmiersprachen ist, dass eine CUDA-Architektur verwendet werden muss. Daher sind alle CUDA C-Programme nur auf CUDA-fähigen Grafikkarten von Nvidia lauffähig. Diese Einschränkung ist jedoch unproblematisch, da Nvidia der Marktführer im Grafikkartenbereich ist und zusammen mit AMD die schnellsten Grafikkarten herstellt. Die Grafikkarten von Nvidia sind nicht teurer als die Konkurrenzprodukte, weshalb diese kleine Einschränkung in Kauf genommen wird. Wer sich nicht auf Grafikkarten von Nvidia festlegen möchte, dem bietet OpenCL eine gute Alternative. Allerdings muss dann eine entsprechende OpenCL Alternative für die zahlreichen CUDA Bibliotheken gefunden werden. Alternativ kann auch die Funktionalität komplett nachprogrammiert werden, was aber einen nicht unerheblichen Zeitfaktor darstellt. Das in der Simulation am meisten genutzte CUDA-Framework ist OptiX und wird im nächsten Abschnitt beschrieben.

5.3 OptiX

Die komplette Berechnung eines Strahles wurde mit Hilfe der Ray-tracing-Bibliothek OptiX (PBD+10) auf die GPU ausgelagert. OptiX basiert auf der oben beschriebenen CUDA-Architektur von Nvidia.

[3] PGI entwickelt zur Zeit einen CUDA-x86 Compiler, der CUDA-Programme für die CPU kompilieren kann.

OptiX ist eine flexibles Framework für GPU basierte Ray-tracing-Anwendungen. Es stellt Datenstrukturen und Algorithmen zur Verfügung die bei Bedarf flexibel ausgetauscht werden können, um Ray-tracing-Anwendungen verschiedenster Art zu entwickeln. Das Framework stellt ein abstraktes Modell für eine Ray-tracing-Anwendung zur Verfügung. Das Modell verwendet programmierbare Schnittstellen für die Erstellung von Strahlen, die Schnittpunktberechnung von Strahlen mit Oberflächen, Materialprogramme und die Erstellung von neuen Strahlen. Durch diese flexible Struktur können mit OptiX verschiedenste Ray-tracing basierte Anwendungen für Grafik, Kollisionserkennung, Schallausbreitung und weitere Bereiche entwickelt werden. Um den Programmieraufwand zu reduzieren, kann auf viele Beispielprogramme zurückgegriffen werden, um typische Algorithmen wie Schnittpunktberechnungen von Dreiecken mit Strahlen zu nutzen. Ein Vorteil dieser flexiblen Struktur ist, dass verschiedene Oberflächenmodelle zeitgleich miteinander verwendet werden können. Für die Beschreibung der menschlichen Geometrie können Mesh-Modelle verwendet werden, da diese am leichtesten zu modellieren sind. Für andere Strukturen, zum Beispiel für Instrumente, kann eine analytische Beschreibung genutzt werden.

Das wesentliche Merkmal einer Ray-tracing-Bibliothek ist die schnelle Berechnung der Schnittpunkte von Strahlen mit beliebigen Objekten. Da diese Schnittpunktberechnung bei komplexen Objekten sehr rechenaufwendig sein kann, werden die im Folgenden beschriebenen Beschleunigungsstrukturen verwendet.

5.3.1 Beschleunigungsstrukturen

Mit Hilfe der Beschleunigungsstrukturen wird vorab getestet, ob ein Objekt von einem Strahl getroffen werden kann. Als Beschleunigungsstrukturen existieren Hierarchien aus Hüllkörpern (BVH; engl. ***B**ounding **V**olume **H**ierarchy*) oder k-d-Bäume [4]. Während BVHs für statische und dynamische Objekte mit verschiedenen Repräsentationen verwendet werden können, spielen k-d-Bäume bei statischen Dreiecksmodellen ihre Stärken aus. Bei statischen Modellen ist hauptsächlich die Ray-tracing-Geschwindigkeit interessant. Bei dynamischen Modellen muss ein gutes Gleichgewicht zwischen Erstellungszeit und der Ray-tracing-Geschwindigkeit gefunden werden. In der Simulation können sowohl statische (z.B. Knochen) als auch dynamische Strukturen (z.B. Muskeln, Organe, Gefäße) vorkommen. Für die BVHs stellt OptiX verschiedene Algorithmen zur Erstellung bereit. Als beste Algorithmen haben sich in verschiedenen Geschwindigkeitsmessungen das Split-BVH (SBVH) für statische Beschleunigungsstrukturen und das MedianBVH für dynamische Strukturen herausgestellt [5].

Die BVHs funktionieren alle nach dem folgenden Prinzip: Für jedes Objekt

[4] Auch hier ist das Framework sehr flexibel und bei Bedarf können eigene Beschleunigungsstrukturen verwendet werden.

[5] Bei Bedarf können auch hier eigene Algorithmen verwendet werden.

wird ein Hüllkörper erstellt, der das Objekt komplett einhüllt. Der Hüllkörper soll möglichst genau die Form des Objektes beschreiben und es soll schnell überprüfbar sein, ob der Hüllkörper von einem Strahl geschnitten wird. Zusätzlich soll er, bei dynamischen Modellen, schnell zu erstellen sein, da er neu berechnet werden muss, sobald sich ein Objekt deformiert. Ein Quader erfüllt diese Eigenschaften insgesamt am besten und wird daher als Hüllkörper verwendet. Andere Hüllkörper, wie zum Beispiel Kugeln, können aber theoretisch auch verwendet werden. Verschiedene Versionen dieser Quader-Hüllkörper sind in Abbildung 5.2 abgebildet. Die Quader können entweder immer in der gleichen Orientierung vorliegen oder sie haben eine beliebige Orientierung. Im ersten Fall bezeichnet man die Quader als *Axis-**A**ligned **B**ounding **B**oxes* (AABB). Diese können sehr schnell erstellt werden, haben jedoch den Nachteil, dass sie bei langen Objekten häufig eine schlechte Passform in Bezug zum Objekt haben. Eine deutlich bessere Passform haben beliebig orientierbare Quader, die ***O**riented **B**ounding **B**oxes* (OBB). Ein OBB ist in der Berechnung deutlich aufwendiger, kann jedoch ein Objekt deutlich besser umhüllen. Trifft der Strahl den Hüllkörper eines Objektes, so werden alle Elemente innerhalb des Hüllkörpers genauer untersucht, ansonsten werden sie verworfen. Diese Optimierungsstrategie ist besonders effizient, wenn möglichst viele kleine Objekte erstellt werden, anstatt weniger großer Objekte. Alternativ können auch SBVHs verwendet werden, welche ein Objekt in mehrere Hüllkörper einteilt. Diese Einteilung ist jedoch sehr rechenaufwendig und wird daher nur für statische Objekte verwendet.

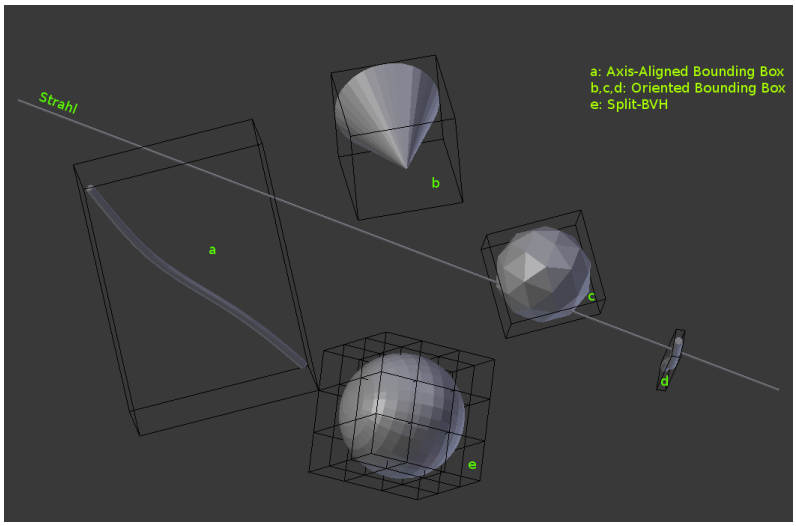


Abbildung 5.2: Beispiel mehrerer BVHs, die auf Schnittpunkte mit einem Strahl überprüft werden.

Die genauere Untersuchung eines Objektes beinhaltet eine Schnittpunktprüfung des Strahles mit jedem Element des Objektes. In der Regel sind diese Elemente

Dreiecke, es können jedoch auch andere Elemente betrachtet werden, deren Schnittpunktalgorithmen jedoch implementiert werden müssen. Im Rahmen dieser Arbeit wurde eine Schnittpunktberechnung von einem Strahl mit einem Kegelstumpf implementiert (siehe 5.3.2). Instrumente wie Katheter und Nadeln können dadurch schnell simuliert werden, ohne vorher ein aufwendiges Mesh-Modell zu erstellen. Die Erstellung der BVHs und die Schnittpunktberechnung für die einzelnen Strahlen laufen in parallelen Prozessen auf der GPU. Dadurch kann eine sehr hohe Simulationsgeschwindigkeit erzielt werden.

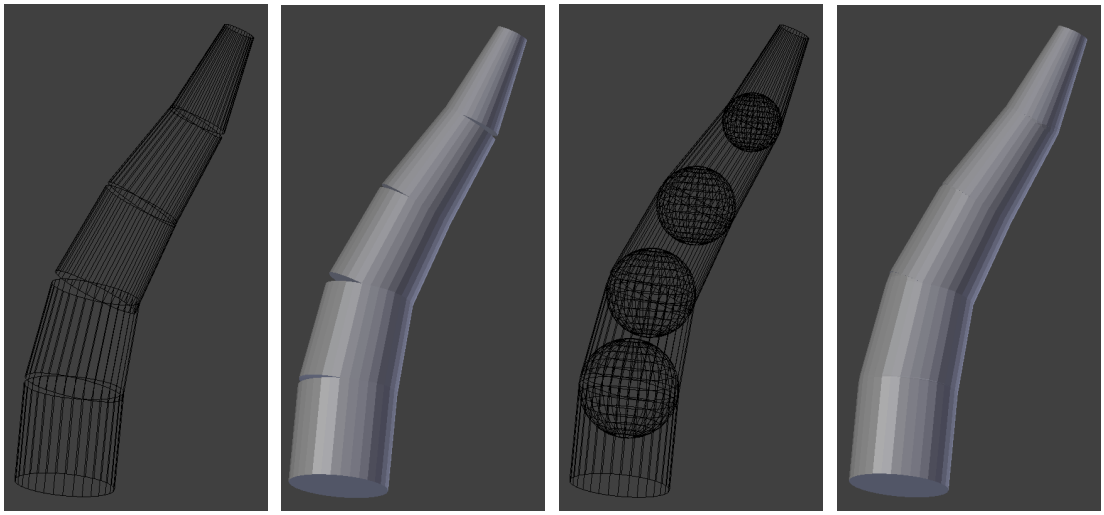
5.3.2 Schnittpunktalgorithmen

OptiX stellt in seinen Beispielprogrammen bereits die wichtigsten Schnittpunktalgorithmen zur Verfügung. Die existierenden Implementierungen für Strahl-Quader, Strahl-Kugel und Strahl-Dreieck wurden leicht abgeändert, um dem Simulationsalgorithmus alle benötigten Informationen zur Verfügung zu stellen. Dies sind die Normale vom getroffenen Objekt am Schnittpunkt und die Koordinaten der 3D-Punkte auf dem Strahl, die vor und hinter dem Schnittpunkt liegen. Für die Simulation nicht benötigte Informationen wie Texturkoordinaten oder die Shadingnormale wurden entfernt, um unnötige Berechnungen zu vermeiden.

Um die Geometrie von Instrumenten in der Simulation zu verwenden, wurde für röhrenförmige Instrumente ein eigener Schnittpunktalgorithmus implementiert. Ein Instrument wird in mehrere Segmente untergliedert. Jedes Segment I besteht aus zwei Koordinaten P_1^I und P_2^I und zwei zugehörigen Radien R_1^I und R_2^I. Die Radien R_2^I und R_1^{I+1} müssen identisch sein. Die Koordinaten mit den Radien werden, wie in Abbildung 5.3a dargestellt, als Kegelstümpfe interpretiert. Falls die Kegelstümpfe nicht alle auf einer Linie liegen, entstehen an den Übergangsstellen der einzelnen Kegelstümpfe gut sichtbare Lücken. Um diese zu schließen, wird an jeder Übergangsstelle eine Linie-Kugel Schnittpunktberechnung durchgeführt (siehe Abbildung 5.3c).

5.3.3 Implementierungsdetails fürs Ray-tracing

OptiX stellt zwei verschiedene Strategien für eine Strahlüberprüfung zur Verfügung, „FirstHit“ und „AnyHit“. In der ersten Variante wird der Anwendercode nur ausgeführt, wenn der Strahl das Objekt mit dem Schnittpunkt S_1 schneidet. S_n bezeichnet hierbei den n-ten Schnittpunkt, in Abhängigkeit zur Entfernung vom Ursprung des Strahles. Alternativ kann der Code bei jedem beliebigen Objekt ausgeführt werden, das den Strahl schneidet. In der ersten Variante stellt OptiX sicher, dass nur der erste Schnittpunkt dem Anwender mitgegeben wird, alle anderen Schnittpunkte werden verworfen. Dies hat den Vorteil, dass nicht alle Objekte aufwendig überprüft werden müssen, sondern es können Objekte ausgeschlossen werden, falls sich diese hinter einem bereits gefundenen Schnittpunkt befinden.



(a) (b) (c) (d)

Abbildung 5.3: Beispiel mehrerer Instrumentensegmente. (a)-(b) zeigen die Segmente ohne Kugeln zwischen den Übergangsstellen, (a)-(b) mit Kugeln.

Dadurch ist die erste Variante etwas schneller bei der Schnittpunktüberprüfung. Bei der zweiten Variante werden jedes Mal, wenn ein Schnittpunkt für ein Objekt gefunden wird, der Schnittpunkt und alle übrigen wichtigen Informationen gespeichert (siehe hierzu 5.3.2). Anschließend wird OptiX mitgeteilt, dass der Schnittpunkt ignoriert und die Schnittpunktberechnung fortgesetzt werden soll. Dies geschieht solange bis kein neuer Schnittpunkt mehr gefunden wird. Sind alle Schnittpunkte festgestellt, dann werden diese in Abhängigkeit zum Abstand vom Ursprung des Strahles aufsteigend sortiert. Diese Sortierung, der zusätzlich benötigte Speicher und die nicht vorhandene Möglichkeit, Objekte aufgrund von früheren Schnittpunkten auszuschließen, machen diese Variante langsamer als die erste. Jedoch bieten sich durch die Kenntnis aller Schnittpunkte auf dem Strahl auch zwei wichtige Vorteile.

Der erste Vorteil ist die Ersparnis von einigen weiteren Schnittpunktberechnungen. Solange der Impedanzunterschied der benachbarten Medien unterhalb eines Schwellwerts liegt, kann man sich die nächste Schnittpunktberechnung ersparen. Ob dies tatsächlich zu einem Geschwindigkeitsvorteil führt hängt von dem Szenarium und dem Schwellwert ab.

Der zweite Vorteil ist, dass die Simulation erheblich stabiler gegenüber nicht idealer Polygonnetze ist. Ein ideales Polygonnetz bezeichnet hierbei ein Mesh, bei dem nicht alle der in Kapitel 6.2 beschriebenen Eigenschaften erfüllt sind. Durch die Kenntnis aller Schnittpunkte kann überprüft werden, ob jeder Eintrittspunkt von dem Mesh auch einen Austrittspunkt besitzt. Tritt ein Fall auf in dem nur ein

Eintritts- oder Austrittspunkt existiert, dann wird dieser Punkt verworfen. Auch bei idealen Mesh-Modellen ist diese zusätzliche Überprüfung sinnvoll, da einige Sonderfälle existieren, bei denen nur ein Schnittpunkt für bestimmte Objekte registriert wird. Diese Sonderfälle sind die Tangenten von Mesh-Modellen bei denen der Strahl nur genau einen Schnittpunkt mit einem Objekt besitzt. Ein weniger offensichtlicher Sonderfall kann bei der Überlappung von zwei Objekten entstehen. Zwar sollte auch eine solche Überlappung von Beginn an ausgeschlossen werden, jedoch lässt sich eine solche leider nicht immer vermeiden. Trifft der Strahl exakt auf die Grenze von einer Überlappung, so wird in der „AnyHit"-Methode nur ein Schnittpunkt registriert und es entstehen unerwünschte Artefakte in der Simulation. In dem Programm wurden beide Strahlüberprüfungsvarianten implementiert. Für die Ergebnisse aus Kapitel 7 wurde jedoch nur die „FirstHit"-Methode verwendet.

5.4 Erweiterungen für eine Brachytherapiesimulation

Die Ultraschallsimulation wird innerhalb eines Brachytherapie-Simulators eingesetzt. Im Zuge dieser Anwendung waren ein paar Modifikationen des ursprünglichen Simulationsalgorithmus erforderlich. Diese wurden als optionale Erweiterungen implementiert und werden in den nächsten beiden Unterabschnitten genauer beschrieben.

5.4.1 Deformation des inneren Gewebes

Bei einer Brachytherapie wird das Gewebeinnere durch eine Nadel deformiert. Die Deformation wird durch eine austauschbare Physik-Engine berechnet. Die aktuelle Implementierung beruht auf einem Finite Element-Modell (FEM). Dieses ist nicht Gegenstand dieser Arbeit und wird daher als *Black Box* betrachtet, welches eine Menge von Punkten M nach M_d abbildet. Die Deformation des Gewebes ist im Ultraschallbild aufgrund der Speckle-Struktur sichtbar. Um die Deformation auch in der Simulation zu berücksichtigen, ist eine Erweiterung des Simulationsalgorithmus erforderlich. Der bisher beschriebene Algorithmus ermöglicht nur eine Deformation der Oberfläche von Objekten, jedoch nicht von der inneren Struktur. Indem man eine Koordinatentransformation auf den Streukörper anwendet, kann man eine Deformationen der inneren Struktur in dem Ultraschallbild simulieren. Dieser Streukörper wird, wie in Kapitel 4.2.2 beschrieben, in einer 3D-Textur gespeichert, die durch $T(X)$ indiziert wird. Die Koordinatentransformation $KT(X)$ wird zwischen den Texturzugriff geschaltet, wodurch sich als neuer Zugriff $T(KT(X))$ ergibt. Die Deformation wird durch ein dreidimensionales Vektorfeld angegeben. Dieses Vektorfeld wird durch einen Quader definiert, der $NxMxO$ Punkte besitzt,

die äquidistant im Inneren des Würfels verteilt sind. Zwischen den Punkten wird trilinear interpoliert, um alle Deformationsvektoren innerhalb des Quaders mit wenigen Punkten definieren zu können. Der Quader ist an den Koordinatenachsen ausgerichtet und kann daher über die zwei Punkte Q_{min} und Q_{max} vollständig beschrieben werden. Alle Punkte des Würfels werden in der 3D-Textur T_d gespeichert. Diese führt eine automatische trilineare Interpolation bei jedem Zugriff aus und bietet darüber hinaus einen sehr effizienten Zugriff. Außerhalb des Quaders ist der Deformationsvektor als Nullvektor definiert. Die Koordinate X muss nur noch für die Texturadressierung angepasst werden. Eine Texturadressierung ist im Bereich 0 bis 1 definiert. Durch Berücksichtigung der beiden Eckpunkte Q_{min} und Q_{max} kann der neue Texturzugriff über Gleichung 5.1 berechnet werden.

$$(X)_d = ((X) - Q_{min})/(Q_{max} - Q_{min}) \tag{5.1}$$

Die Koordinatentransformation KT wird dann über eine Addition des Texturzugriffs und der ursprünglichen Koordinate berechnet.

$$KT(X) = T_d((X)_d) + (X) \tag{5.2}$$

Die Deformation erscheint umso realistischer, je größer die 3D-Textur T_d ist. Allerdings wirkt sich eine zu große Textur negativ auf die Simulationszeit aus. Die Aktualisierung der Textur selbst ist nicht zeitintensiv. Der zeitkritische Punkt ist die Aktualisierung der Deformationspunkte durch die Physik-Engine. Um diese etwas zu entlasten, ist es sinnvoll die Größe der Textur zu minimieren und nur die Deformationspunkte zu erfassen, die auch tatsächlich relevant für die Simulation sind. Dadurch reduziert sich der Simulationsaufwand der Physik-Engine erheblich. Ein Nachteil dieser Methode ist, dass die Lage der Deformationspunkte neu berechnet werden muss, falls sich der Schallkopf signifikant bewegt. Diese Neuberechnung ist bei der aktuellen FEM-Implementierung sehr zeitintensiv. Daher wird sie in einen eigenen Thread ausgelagert. Die Physik-Engine nutzt die alten Deformationspunkte bis die neuen zur Verfügung stehen. Dies gewährleistet, dass die Ultraschallsimulation in definierten Abständen neue Daten von der Physik-Engine erhält. Da sich bei der Brachytherapie die Lage des Schallkopfes nur sehr selten und in kleinen Abständen verschiebt, ist dieses Verfahren hier sehr gut geeignet. Bei häufigen und schnellen Neupositionierungen des Schallkopfes, wie sie zum Beispiel bei Abdominaluntersuchungen auftreten, kann dieses Verfahren nicht verwendet werden. Hier wäre eine Einteilung des gesamten Simulationsbereiches in mehrere sich überlappende Quader sinnvoll. Es müsste dann bestimmt werden welcher der Quader das Ultraschallsimulationsgebiet am besten abdeckt. Dieser Quader würde dann in Physik- und Ultraschallsimulation verwendet werden, während alle anderen Quader deaktiviert bleiben könnten. Der Nachteil dieser Methode wäre ein erheblich höherer Speicherbedarf.

5.4.2 Materialtypen

In der Ultraschallsimulation können unterschiedliche Materialtypen verwendet werden. Der Standardmaterialtyp ist das Material von dem bisher ausgegangen wurde. Alle Streukörper bleiben an der gleichen Stelle, es sei denn das Material wird durch einen externen Einfluss wie eine Deformation beeinflusst.

Ein weiterer verwendeter Materialtyp ist Blut. Dieses fließt innerhalb der Gefäße und die Streukörper ändern sich mit jedem Zeitschritt. Daher muss die Rauschstruktur nicht gespeichert werden und anstatt des Streukörpers $T(x, y, z)$ wird eine normalverteilte Zufallsfunktion genommen.

Im Simulator werden weiterhin die beiden Materialtypen Seeds und Instrumente verwendet. Die in der Brachytherapie verwendeten Seeds sind nach dem gleichen Prinzip aufgebaut. Die kleinen Stifte, mit einer Länge von 25 mm bis 80 mm und einem Durchmesser von ungefähr einem Millimeter, enthalten in ihrem Inneren ein radioaktives Element. In Deutschland wird in der Regel IOD^{125} verwendet, das in einem Silberkern gebunden ist. Die äußere Schicht besteht üblicherweise aus einer sehr dünnen Titanschicht, welche eine sehr hohe Schallimpedanz aufweist. Durch diese hohe Schallimpedanz und den kleinen Abstand entsteht in einem Ultraschallbild häufig ein Kometenschweifartefakt (siehe Kapitel 2.3.6). Der Ultraschallstrahl wird sehr oft innerhalb des Seeds reflektiert und für jede Reflexion wird ein neuer Strahl erstellt. Die gleichen Beobachtungen gelten auch für die in der Brachytherapie verwendete Hohlnadel. Die Anzahl der benötigten Schnittpunktberechnungen steigt durch diese häufigen Reflexionen deutlich an. Die Anzahl der Schnittpunktberechnungen beeinflusst die benötigte Simulationszeit um einen exponentiellen Faktor. Das Artefakt kann zwar mit der bisherigen Methode gut simuliert werden, jedoch kosten diese Berechnungen sehr viel unnötige Rechenzeit. Daher wird für die Seeds und Nadeln ein eigener Materialtyp verwendet, der das Artefakt künstlich in das Ultraschallbild einbaut. Hierzu wird der Abstand d_r zwischen den ersten beiden Reflexionen gemessen. Zu jedem Abtastpunkt, der ein Vielfaches von d_r ist, und sich hinter dem Material befindet wird ein Wert addiert. Dieser hängt von der Intensität des Strahles am Abtastpunkt und von dem Austrittswinkel des Strahles aus dem Material ab.

5.5 Koordinatentransformation

Die kompletten Simulationsergebnisse liegen linear im Speicher. Alle Abtastpunkte eines Strahles, im Folgenden Scanline S genannt, liegen hintereinander im Speicher, gefolgt von der Scanline des nächsten Strahles. Durch eine Koordinatentransformation wird die berechnete Rückstreuung analog zur Ausbreitungsrichtung der Schallstrahlen dargestellt.

Da bei einem Linear-Array Schallkopf die Strahlen parallel zueinander ausgesendet werden, entfällt hier eine aufwendige Koordinatentransformation. Es ist

ausreichend, das Ultraschallbild über eine bilineare Interpolation der einzelnen Scanlinien zu erzeugen. Diese einfache bilineare Interpolation wird an dieser Stelle nicht näher beschrieben.

Für alle anderen Schallköpfe ist die Koordinatentransformation aufwendiger. Die Strahlen sind entlang eines Kreisbogens angeordnet. Alle Strahlen treffen sich, wie in Abbildung 5.4 zu erkennen, in einem imaginären Ursprung O. Sie unterscheiden sich nur um den äquidistanten Winkel φ. Den einzelnen Abtastpunkten eines Strahles werden die Polarkoordinaten r für den Abstand des Punktes zum Ursprung und α für den Winkel zugewiesen.

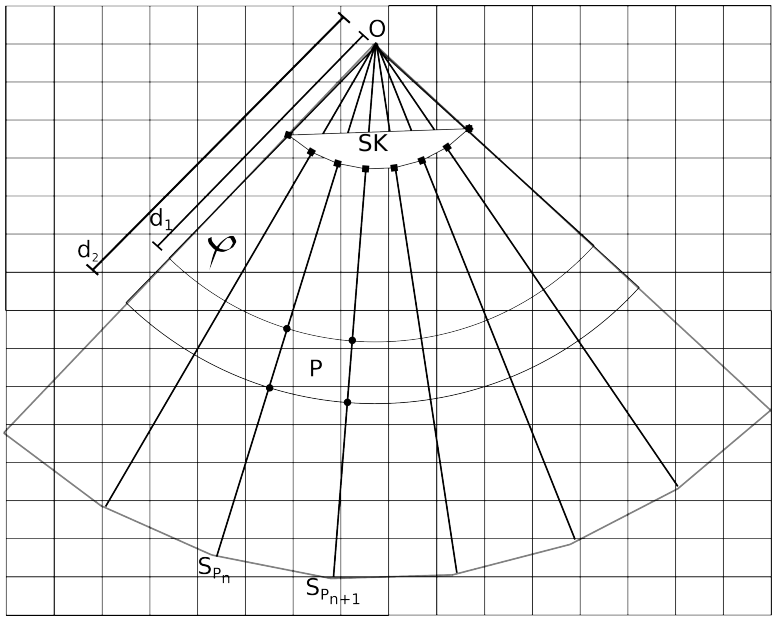


Abbildung 5.4: Beispiel einer Koordinatentransformation anhand eines Konvex-Array Schallkopfes (SK).

Die Bestimmung des Grauwerts für einen Pixel $P(x, y)$ erfolgt über eine Umwandlung der kartesischen Koordinaten in die Polarkoordinaten $P(r, \varphi)$. Die Berechnung von r und φ erfolgt mit der Formel 5.3. In CUDA C und C++ existiert für diese Fallunterscheidungen die Funktion $atan2(y, x)$.

$$r = \sqrt{x^2 + y^2} \qquad \varphi = \begin{cases} \arctan \frac{y}{x} & \text{für } x > 0 \\ \arctan \frac{y}{x} + \pi & \text{für } x < 0,\ y \geq 0 \\ \arctan \frac{y}{x} - \pi & \text{für } x < 0,\ y < 0 \\ +\pi/2 & \text{für } x = 0,\ y > 0 \\ -\pi/2 & \text{für } x = 0,\ y < 0 \end{cases} \tag{5.3}$$

Für die Transformation werden die beiden Scanlines links S_{P_n} und rechts von

dem Pixel $S_{P_{n+1}}$ ausgewählt. Für beide wird jeweils ein oberer $S(d_2)$ und unterer Abtastpunkt $S(d_1)$, entsprechend der r-Koordinate des Punktes $P(r, \varphi)$ gewählt. Zwischen diesen vier Punkten wird nun eine bilineare Interpolation durchgeführt, um den Grauwert des Punktes zu ermitteln. Dieser Vorgang wird für jeden darzustellenden Pixel durchgeführt. Befindet sich der Pixel außerhalb des Schallbereiches, wird ihm ein definierter Grauwert zugewiesen. Für die Transformation des IVUS Bildes wird zusätzlich die Fallunterscheidung durchgeführt, ob sich der Pixel im oberen oder im unteren Teil des Kreises befindet. Die Algorithmen sind in CUDA C implementiert, daher ist die bilineare Interpolation über einen Texturzugriff realisiert.

5.6 Performanzoptimierung

Da die Simulation interaktiv ablaufen soll, muss der Programmcode gut optimiert sein, um eine möglichst hohe Bildwiederholungsrate zu erreichen. Neben der bereits beschriebenen Verwendung der Grafikkarte wurden noch mehrere weitere Schritte unternommen, um den Programmcode zu optimieren. Die wichtigste Optimierung betrifft die Algorithmen selbst. Die von OptiX verwendeten Beschleunigungsstrukturen wurden bereits in Abschnitt 5.3.1 beschrieben. Außerdem werden für das Ray-tracing nur die Strahlen berechnet, die noch vom Schallkopf empfangen werden können (engl. *Early-Ray-Termination*). Der CPU-Code ist in mehrere Threads unterteilt, welche im nächsten Abschnitt genauer beschrieben werden. Am Schluss wurden die zeitkritischsten Funktionen auf weitere Optimierungsmöglichkeiten untersucht.

5.6.1 Threads und Synchronisation

Um die verschiedenen Teile der Simulation möglichst unabhängig voneinander laufen zu lassen, werden sie in verschiedene Threads aufgeteilt.

Die Ultraschallsimulation und die komplette grafische Benutzeroberfläche (*GUI*; engl. ***G**raphical **U**ser **I**nterface*) werden in dem *Simulation-Thread* abgearbeitet. Bei Bedarf könnten diese noch voneinander entkoppelt werden. Da die GUI sehr wenig Rechenzeit benötigt lohnt sich der Mehraufwand für eine Entkopplung derzeit nicht.

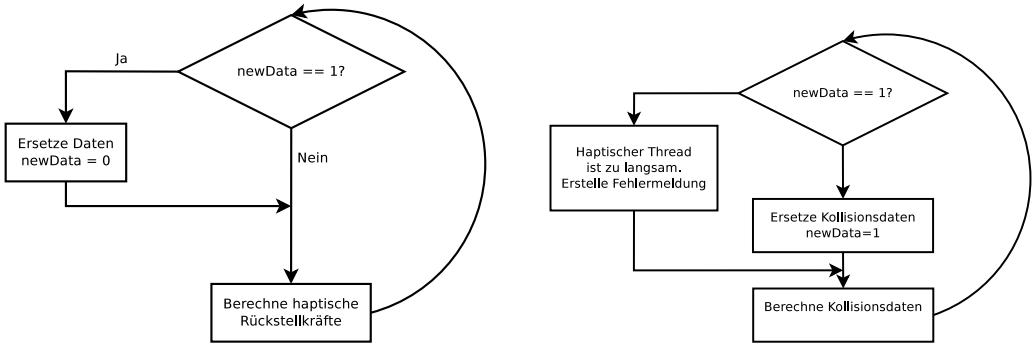
Der Algorithmus zur Steuerung des haptischen Gerätes läuft in einem *Haptik-Thread*. Zwar ist der Rechenaufwand auch hier sehr klein, jedoch müssen dem Gerät die Rückstellkräfte mit einer sehr hohen Frequenz (mehr als 1000 Updates pro Sekunde) übermittelt werden. Wird diese hohe Frequenz nicht eingehalten, verspürt der Anwender ein unerwünschtes Ruckeln.

Der nächste Thread ist für die Physiksimulation reserviert. Da auch hier eine hohe Simulationsgeschwindigkeit benötigt wird, läuft der *Physik-Thread* unabhängig von den anderen Threads. Bei einem höheren Rechenbedarf können auch

mehrere Threads verwendet werden. Der *Kollision-Thread* ist für die Kollisionsberechnung von dem Schallkopf mit der restlichen Geometrie zuständig. Er kann im *Physik-Thread* laufen, bei Bedarf jedoch auch in einen weiteren eigenständigen Thread ausgelagert werden.

Folgende Threads müssen miteinander synchronisiert werden, da Abhängigkeiten vorliegen. Der *Haptik-Thread* benötigt einen Schnittpunkt, einen Abstand sowie eine Normale vom *Kollision-Thread*. Der *Simulation-Thread* benötigt die aktuelle Geometrie vom *Physik-Thread*. Die Threads synchronisieren sich nicht, wie üblich, über Mutexe, sondern über normale Variablen. Dies ist möglich, da die spezifischen Aktualisierungseigenschaften jedes Threads genau berücksichtigt werden und gewährleisten, dass keine unnötigen Synchronisationszeiten entstehen.

Der *Haptik-Thread* holt sich neue Kollisionsdaten und setzt die Variable *newData* auf null, falls über die gleiche Variable angezeigt wird, dass neue Daten vorhanden sind. Danach arbeitet er seinen Algorithmus ab. Der *Kollision-Thread* kopiert alle Kollisionsdaten in einen speziell reservierten Datenbereich. Danach setzt er die Variable *newData* auf eins und beginnt mit der Berechnung der neuen Kollisionsdaten. Sollte die Variable *newData* nach den Berechnungen der Kollisionsdaten noch gesetzt sein, so wird eine Fehlermeldung ausgegeben, um anzuzeigen, dass der *Haptik-Thread* zu langsam arbeitet. Da der *Haptik-Thread* mit ungefähr 1000 Aktualisierungen pro Sekunde arbeitet und der *Kollision-Thread* mit weniger als 100 Aktualisierungen, tritt dieser Fall aber i.d.R. nicht auf. Die Funktionsweise des Algorithmus wird in Diagramm 5.5 veranschaulicht.

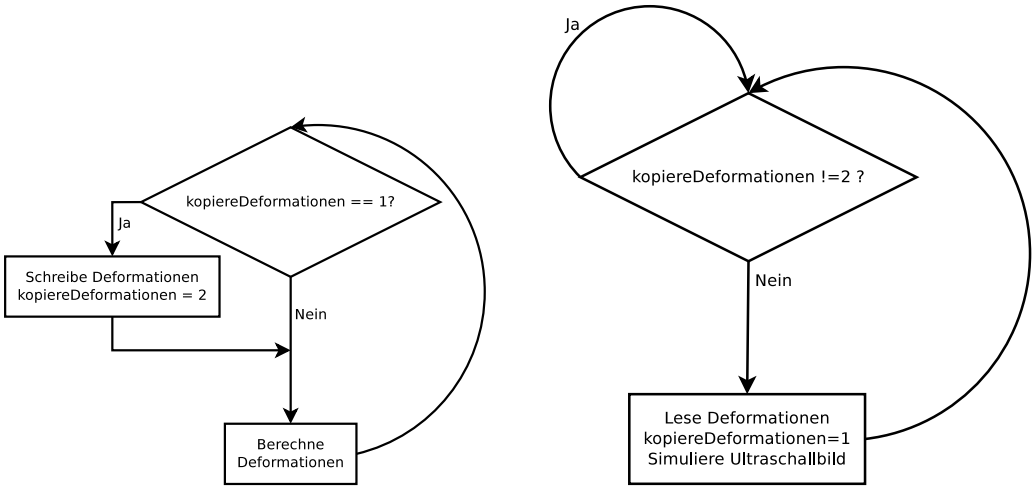


(a) Haptik-Thread (b) Kollision-Thread

Abbildung 5.5: Flussdiagramm der haptischen Synchronisation.

Die Synchronisation zwischen *Physik-Thread* und *Deformation-Thread* arbeitet nach einem ähnlichen Prinzip. Der *Simulation-Thread* fordert die aktuellen Deformationsdaten vom *Deformation-Thread* an, indem die Variable $kopiereDeformationen$ auf eins gesetzt wird. Anschließend simuliert er das Ultraschallbild. Innerhalb dieser Zeit kopiert der *Physik-Thread* die Deformationsdaten und setzt $kopiereDeformationen$ auf zwei, um anzuzeigen, dass die Daten bereitstehen. Sobald der *Simulation-Thread* das Ultraschallbild simuliert hat, geht

er in den Schlafmodus und wartet mit der weiteren Ausführung, bis die Daten bereitstehen. Im Idealfall ist dies sofort nach der Simulation des Ultraschallbildes der Fall, da ein Durchlauf des *Physik-Threads* schneller sein sollte als die Simulation eines Ultraschallbildes. Sollte der *Physik-Thread* langsamer sein als die Ultraschallsimulation, dann benötigt die Ultraschallsimulation die gleiche Zeit zur Simulation wie der *Physik-Thread*. Dies könnte man durch die Wahl eines anderen Synchronisationsalgorithmus verhindern. Bei einer langsamen Physik-Simulation wirkt die komplette Simulation aber ohnehin nicht flüssig, daher ist es empfehlenswert die Geschwindigkeit der Physik-Simulation zu verbessern, anstatt einen anderen Synchronisation-Algorithmus zu verwenden. Die Funktionsweise des Algorithmus wird in Diagramm 5.5 veranschaulicht.



(a) Physik-Thread (b) Simulation-Thread

Abbildung 5.6: Flussdiagramm der Physik Synchronisation.

5.6.2 Funktions- und Stabilitätsoptimierungen

Der in C++ programmierte Code wurde mit dem Open-Source-Programm valgrind (www.valgrind.org) optimiert. Zuerst wurde überprüft, ob der Code Speicherzugriffsfehler verursacht. Diese Speicherfehler können uninitialisierte Variablen, Zugriffe auf nicht registriere Speicherbereiche, fehlende oder mehrfache Speicherfreigaben sein. Alle durch das Programm entdeckten Speicherfehler wurden behoben. Anschließend wurden alle rechenintensiven Funktionen optimiert.

Der in CUDA C programmierte Code wurde vollständig optimiert. Auf die Verwendung eines Profilers wurde verzichtet, da der auf OptiX basierende Code nicht gut ausgewertet werden kann. Die Benutzung eines Profilers bringt an dieser Stelle aber auch keine Vorteile, da die zeitkritischen Programmteile bereits bekannt sind. Jeder Code, der auf der GPU läuft und keine Initialisierungsfunktion ist, ist

zeitkritisch. Es wurde besonders auf eine Minimierung des verwendeten Speichers geachtet, da dieser auf der GPU nur begrenzt zur Verfügung steht, beziehungsweise die Anzahl der gleichzeitig ausführbaren Threads einschränkt. Außerdem wurde darauf geachtet, nur Gleitkommazahlen mit einfacher Genauigkeit und nur die jeweils hierfür vorgesehenen Funktionen zu verwenden.

6 Modellerzeugung

6.1 2D-Modelle

Die 2D-Modelle basieren auf einem eigenen Format, das in Abschnitt 4.1.1 beschrieben wird. Um passende Modelle zu erstellen wird ein selbst entwickelter Editor verwendet. In diesem Editor können Modelle erstellt und modifiziert und die Ultraschalleigenschaften jedes Objektes angepasst werden. Der Editor kann Gefäßbäume in einem speziellen Format importieren und daraus fertige Modelle erstellen. Bifurkationen werden zur Zeit nicht automatisch generiert. Sie können bei Bedarf vom Nutzer manuell eingearbeitet werden.

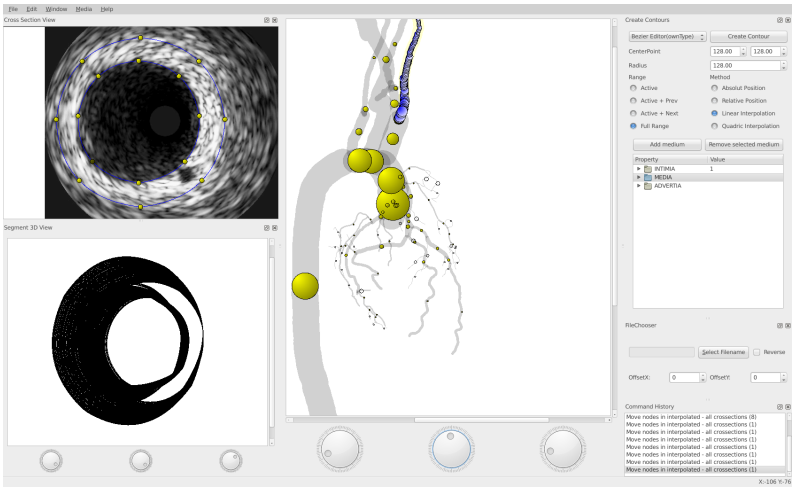


Abbildung 6.1: Screenshot des IVUSEditors

Um das Editieren der Modelle zu vereinfachen, wurden verschiedene Operationen implementiert. Der Gefäßbaum und Teilsegmente können als dreidimensionale Objekte angezeigt werden. Segmente können als Ganzes bearbeitet werden, da das Verschieben eines oder mehrerer Knotenpunkte über lineare oder quadratische Interpolation Auswirkungen auf das gewählte Teilsegment hat. Das *Command-Pattern* wird genutzt, um Operationen rückgängig machen zu können. Gängige Transformations-Operationen wie skalieren, rotieren und verschieben von Knotenpunkten können verwendet werden. Als Hintergrundbilder können das simulierte Bild oder Bildsequenzen von echten Ultraschallbildern angezeigt werden. Das

passende Ultraschallbild wird hierbei automatisch in Abhängigkeit vom gesetzten Startpunkt ausgewählt.

6.2 3D-Modelle

Mesh-Modelle können von kommerziellen Anbietern erworben oder selbst erstellt werden. Gekaufte Mesh-Modelle bringen einige Nachteile mit sich. Genaue anatomische Modelle sind sehr teuer und dürfen häufig nur auf einem Rechner verwendet werden. Für die Verwendung des Simulators auf mehreren Rechnern müssen also mehrere Lizenzen erworben werben. Es muss geprüft werden, ob die Modelle anatomisch korrekt sind und ob genügend Details vorhanden sind. Die Modelle sind im Allgemeinen auf visuelle Darstellung optimiert und ohne Nachbearbeitung nicht für Simulationen geeignet. Die in Abschnitt 6.2.1 beschrieben Eigenschaften eines Mesh-Modells sind i.d.R. nicht erfüllt. Die Mesh-Modelle sind häufig nicht 2-mannigfaltig, nicht geschlossen und es existieren Selbstüberschneidungen. Ein weiterer Nachteil von gekauften Mesh-Modellen ist, dass man auf ein Modell festgelegt ist. Dieses kann zwar verändert werden, um andere Untersuchungsfälle zu simulieren, jedoch ist es nicht möglich eine patientenspezifische Simulation zu erstellen.

Eine Alternative gegenüber kommerziellen Mesh-Modellen ist es, eigene Modelle aus vorhandenen CT- oder MRT-Volumen zu generieren. Die Volumen müssen segmentiert und klassifiziert werden und anschließend können aus den Voxeldaten Oberflächenmodelle erstellt werden. Diagnostisch wichtige Strukturen, die aufgrund der im Vergleich zum Ultraschall geringeren Auflösung von CT bzw. MRT nicht vorhanden sind, müssen komplett manuell erstellt werden. Der Königsweg ist die vollautomatische Segmentierung und Klassifizierung von hochauflösenden Datensätzen, damit patientenspezifische Simulationen ohne Aufwand erstellt werden können. Im Rahmen dieser Arbeit war die jedoch zeitlich nicht möglich, daher werden verschiedene Wege genutzt, um für die Simulation geeignete Modelle zu erzeugen. Für die Modellierung der beiden Phantom-Modelle werden CT-Volumen verwendet. Mit den im medizinischen Bildverarbeitungsframework „MeVisLab“ enthaltenen Filtern wird eine Bildverarbeitungspipeline (Bildverarbeitungskette) generiert, um die Modelle zu erstellen. Bildverarbeitungsmethoden wie Rauschentfernung (*Median Filter*), Glättung (*Gaussian Smoothing*) und Segmentierung (*Connected Threshold, Region Growing*) werden aneinandergereiht, um segmentierte Objekte zu erzeugen. Diese werden anschließend manuell klassifiziert. Aus diesen Voxelobjekten werden durch Oberflächengenerierungsalgorithmen (*Marching Cube*), Polygonglättungsfilter (*Laplacian Surface Smoothing*) und Polygonreduzierungsfilter (*Quadric Error Metrics*) für die Simulation geeignete Mesh-Modelle generiert. Ein Beispiel einer solchen Bildverarbeitungskette, mit dem das 3D-Modell des BAT-Phantoms erstellt wurde, ist in Abbildung 6.2 zu sehen.

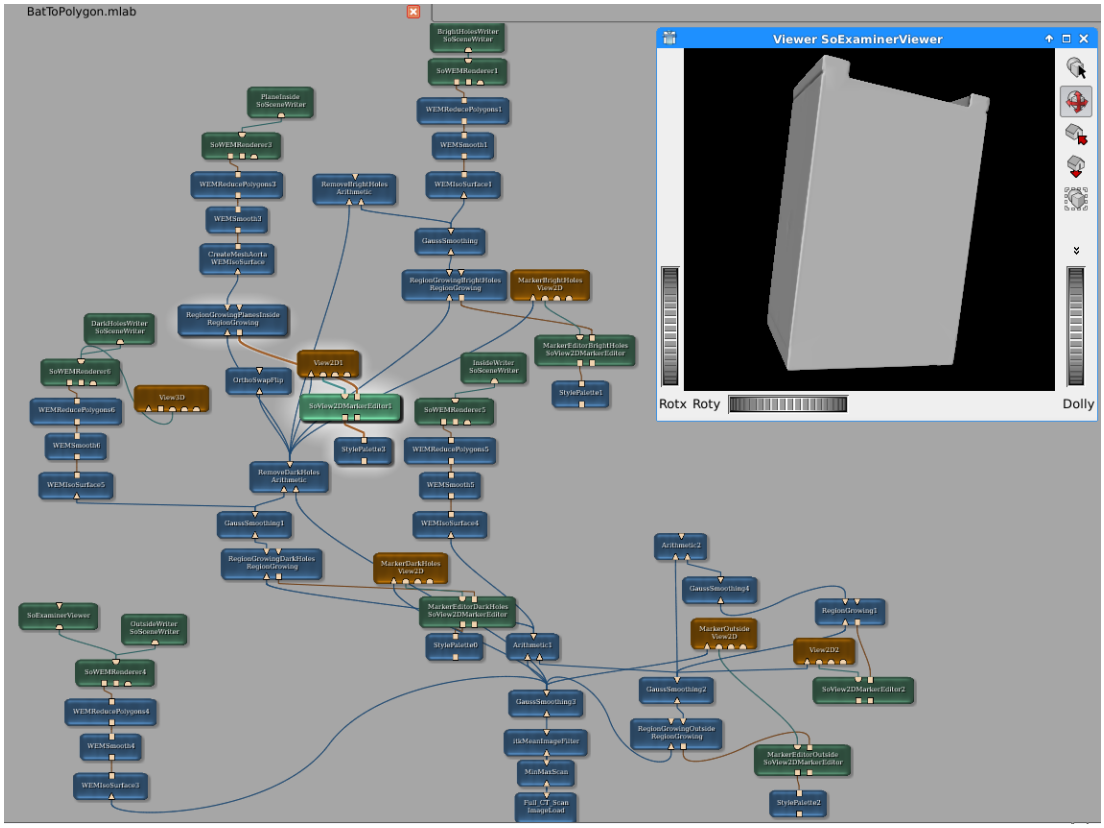


Abbildung 6.2: Bildverarbeitungskette von hintereinandergeschalteten Filtern in „MeVisLab“ zur Segmentierung des BAT-Phantoms

6.2.1 Nachbearbeitung von Mesh-Modellen

Als anatomisches Modell wird ein kommerzielles Mesh-Modell genutzt. Da auch dieses Gitternetz nicht ohne Nachbearbeitung für die Simulation nutzbar ist, wurden einige Programme und Skripte entwickelt, um das Mesh semiautomatisch nachzubearbeiten.

Alle Mesh-Modelle liegen im Wavefront-Format vor, bei dem es sich um ein ASCII-Format handelt. Die Beschreibung wird zeilenweise eingelesen, wobei der erste Buchstabe jeweils die Art der Beschreibung angibt. Im Folgenden werden nur die elementarsten Beschreibungen aufgelistet, die vom Simulator verwendet werden. Alle anderen Beschreibungen wie zum Beispiel Normale, Texturkoordinaten oder Materialeinstellungen werden vom Simulator ignoriert.

- # - Die komplette Zeile ist nur ein Kommentar und wird ignoriert.

- v x y z - Die Zeile beschreibt einen Vertex mit den Koordinaten x y und z. Die Koordinaten werden jeweils in Dezimalschreibweise angegeben.

- f v1 v2 v3 . . . - Die Zeile beschreibt ein Polygon aus mindestens drei Punkten.

Zuerst werden alle Punkte der Oberfläche aufgelistet. Jeder Vertex wird automatisch nummeriert, beginnend mit der Nummer eins. Diese Vertexnummerierung wird genutzt, um anschließend alle Polygone zu beschreiben. Die Vertexnummerierung beschreibt welche Vertices miteinander verbunden werden, um ein Polygon zu erzeugen. Die letzte Vertexnummerierung ist immer identisch mit der ersten Nummerierung und wird daher weggelassen. Ein Würfel könnte also folgendermaßen beschrieben werden:

```
1  # Blender v2.62
2  # www.blender.org
3  v 1.0 -1.0 -1.0
4  v 1.0 -1.0 1.0
5  v -1.0 -1.0 1.0
6  v -1.0 -1.0 -1.0
7  v 1.0 1.0 -1.0
8  v 1.0 1.0 1.0
9  v -1.0 1.0 1.0
10 v -1.0 1.0 -1.0
11 v 1.0 -1.0 1.0
12 v 1.0 -1.0 1.0
13 v 1.0 -1.0 -1.0
14 v 1.0 -1.0 -1.0
15 v -1.0 -1.0 -1.0
16 v -1.0 -1.0 -1.0
17 v -1.0 -1.0 1.0
18 v -1.0 -1.0 1.0
19 v -1.0 1.0 1.0
20 v -1.0 1.0 1.0
21 v 1.0 1.0 1.0
22 v -1.0 1.0 -1.0
23 v -1.0 1.0 -1.0
24 v 1.0 1.0 -1.0
25 v 1.0 1.0 1.0
26 v 1.0 1.0 -1.0
27 s off
28 f 1 2 3 4
29 f 5 8 7 6
30 f 12 24 23 10
31 f 9 19 18 16
32 f 15 17 20 14
33 f 22 11 13 21
```

In dem Beispiel sind einige Punkte doppelt vorhanden. Diese Duplikate werden häufig von 3D-Grafikprogrammen erzeugt, um einem Vertex mehrere Normale übergeben zu können. Für den Ray-tracing-Algorithmus sind diese doppelten Vertices störend, daher werden sie direkt beim Einlesen der Mesh-Modelle entfernt. Dies geschieht, indem jeder Vertex mit allen anderen verglichen wird. Sind alle drei Punkte identisch, dann liegt ein gleicher Vertex vor und einer der beiden Vertices wird entfernt. Wird der gelöschte Vertex in einem Polygon verwendet, dann wird die Nummerierung des gelöschten Vertex mit der Nummerierung des Doppelgängers ersetzt.

In dem Simulator werden keine allgemeinen Polygone verwendet, sondern, wie in der 3D-Programmierung üblich, Dreiecke. Jedes Polygon lässt sich mit Hilfe des *Ear-Clipping* Algorithmus in mehrere Dreieck umwandeln. Dieser Schritt wird automatisch beim Laden des 3D-Modells ausgeführt. Daher kann die Oberflächenbeschreibung auch in Form von Polygonen vorliegen.

Ein Mesh-Modell muss, aufgrund des Ray-tracing-Algorithmus, ein definiertes Inneres und Äußeres haben und darf keinen Rand besitzen. Wird ein Mesh-Modell von einem Strahl getroffen, so muss klar definiert sein, ob der Strahl die

äußere Seite eines Objektes getroffen hat oder die innere Seite. Etwas formaler ausgedrückt bedeuten diese Eigenschaften, dass das Modell geschlossen sein muss und 2-mannigfaltig. Ein Mesh ist geschlossen, wenn es keinen Rand besitzt, und 2-mannigfaltig, wenn folgende Eigenschaften erfüllt sind:

- *Local Disc Property*: Um jeden beliebigen Punkt auf dem Mesh lässt sich immer eine ϵ-Umgebung (= Kugel mit Radius $\epsilon > 0$) finden, so dass die Schnittmenge aus ϵ-Umgebung und Mesh homöomorph zu einer planaren Scheibe ist. Sofern es sich um einen Punkt auf dem Rand handelt, kann die Schnittmenge auch homöomorph zu einem Halbkreis sein.

- *Edge Ordering Property*: Die Nachbarvertices eines jeden Vertex lassen sich eindeutig im Uhrzeigersinn aufzählen.

- *Face Count Property*: Jede innere Kante hat genau zwei adjazente Faces. Jede Kante, die am Rand liegt, hat genau ein adjazentes Face.

Es gibt eine einfache Regel mit der die 2-Mannigfaltigkeit und die Geschlossenheit eines Mesh-Modells leicht überprüft werden kann. Jede Kante des Modells muss genau von zwei Dreiecken genutzt werden.

Eine weitere Eigenschaft, welche die Mesh-Modelle erfüllen müssen ist, dass keine Selbstüberschneidungen vorliegen. Dies wird überprüft, indem eine Schnittpunktberechnung zwischen allen Dreiecken durchgeführt wird.

Wenn möglich, sollten auch keine Überschneidungen mit anderen Objekten vorliegen. Bei Strukturen, die den kompletten Körper durchlaufen wie Nerven und Blutgefäße, ist dies nicht möglich. Bei allen anderen anatomischen Strukturen sollte jedoch darauf geachtet werden. Dies geschieht, indem alle Dreiecke eines Objektes mit allen Dreiecken eines anderen Objektes auf gemeinsame Schnittpunkte überprüft werden.

Alle Eigenschaften werden automatisch mit Hilfe eines selbstentwickelten Programms überprüft. Wird eine Eigenschaft gefunden, die nicht erfüllt ist, so wird versucht diese automatisch mit Hilfe von automatisierten Aufrufen der 3D-Modellierungssoftware *Blender* zu beheben. Sollte dies nicht möglich sein, können die Mesh-Modelle automatisch in *Blender* geladen werden und die problematischen Stellen werden hervorgehoben, damit der Anwender diese schnell bereinigen kann.

6.3 Parameterschätzung

Um einen bestimmten Ultraschallkopf zu simulieren, müssen viele Parameter gesetzt werden. Theoretisch lassen sich alle Parameter bei genauer Kenntnis des zu simulierenden Ultraschallkopfes berechnen. Da ein bestimmter medizinischer Ultraschallkopf aber in der Regel eine *Black Box* ist und nur der Hersteller alle

zur Berechnung notwendigen Parameter kennt, werden die meisten benötigten Parameter geschätzt. Da die Schätzung aller Einstellungen sehr aufwendig ist, wurde hierfür eine semiautomatische Methode entwickelt.

Die Software wurde in Matlab implementiert und verbindet sich über eine TCP/IP Schnittstelle mit dem Ultraschallsimulator. Diesem werden alle zu schätzenden Parameter übergeben und als Ergebnis erhält die Parameterschätzmethode ein simuliertes Bild. Dieses Bild wird mit einem echten Ultraschallbild verglichen. Auf Basis dieses Vergleiches werden dann die Parameter in einem iterativen Prozess weiter optimiert. Die Parameterschätzung wird in einer festen Reihenfolge ausgeführt, da es rechnerisch zu aufwendig wäre, alle Parameter in Abhängigkeit zu den anderen Parametern zu simulieren.

Als erste und einfachste zu bestimmende Parameter werden die geometrischen Ausmaße des Bildes berechnet. Darunter fällt zum einen die Reichweite des Ultraschalls als auch die Anordnung der Ultraschallwandler bei einem konvexen Scan bzw. der maximale Öffnungswinkel bei einem Sektorscan.

Die nächsten zu schätzenden Parameter des Ultraschallkopfes betreffen die Auflösung des Ultraschallkopfes. Hierfür werden die Speckles in einem manuell gewähltem Bereich des echten Ultraschallbildes analysiert. Zuerst werden die Speckles durch einen von Stippel et al. (SPL02) entwickelten Algorithmus lokalisiert. Dieser verwendet als Kriterium für die Lokalisierung die Tatsache, dass die Speckle-Struktur heller als ihre Umgebung ist. Der Algorithmus funktioniert folgendermaßen:

1. Finde alle lokalen Maxima, deren Grauwerte G(x,y) einen Schwellenwert überschreiten. Speichere diese in einen Array M.

2. Nehme das größte verbliebene lokale Maximum an der Stelle (i,j). Dieser Pixel mit Grauwert G(i,j) ist der Startpunkt, um den der Speckle entwickelt wird. Ein benachbarter Pixel an der Stelle (m, n) gehört zu dem Speckle wenn:
   - (m, n) mit (i, j) verbunden ist.
   - (m, n) zu keinem anderen Speckle oder der Nachbarschaft eines Speckles gehört.
   - $G(i,j) - G(m,n) > \tau$, wobei der Grenzwert τ manuell gesetzt wird.

3. Entferne alle lokale Maxima, die zu dem entwickelten Speckle gehören oder sich in der direkten Umgebung des Speckles befinden.

4. Wenn M nicht leer ist, gehe zu Punkt 2.

Nach der Lokalisation aller Speckles wird die durchschnittliche Breite S_b und Höhe S_h der Speckles berechnet. Basierend auf diesen Werten wird der minimale Wert für den Parameter *ScaleScatter* berechnet.

Eine weitere Möglichkeit, um das Simulationsergebnis mit dem echten Ultraschallbild zu vergleichen, bietet ein Vergleich von zwei Bildausschnitten durch Methoden aus der Ordnungsstatistik (*Second Order Statistic*). Grauwertmatrizen (*GLCM*; engl. ***g**ray **l**evel **co**-occurence **m**atrix*) geben die Häufigkeit von benachbarten Grauwerten an. Die Nachbarschaft lässt sich durch verschiedene Richtungen bestimmen. In der Regel werden vier Matrizen gebildet, welche als Nachbarschaftsfunktion jeweils einen der vier Winkel 0°, 45°, 90° und 135° verwenden. Die Grauwertmatrizen wurden erstmals von Haralick (Har79) für die statistische Bildanalyse verwendet und sind folgendermaßen definiert,

$$C\,(i,j) = \sum_{p=1}^{n}\sum_{q=1}^{m}\begin{cases}1, & if\ I(p,q) = i\ and\ I(p+\triangle x, q,\ p+\triangle y) = j\\ 0, & otherwise\end{cases} \tag{6.1}$$

wobei nxm die Größe des zu untersuchenden Bildes bzw. Bildbereiches angibt. Die Größe der Grauwertmatrix ergibt sich aus dem verwendeten Grauwertbereich. Möchte man die Matrix verkleinern, so können mehrere zusammenhängende Grauwerte zu einem Grauwert zusammengefasst werden: $G' = \frac{G}{a}a, G, G' \in \mathbb{N}$. Für a werden i.d.R. Vielfache von zwei verwendet.

Für jede dieser Matrizen können verschiedene Eigenschaften ausgewertet werden, die von Haralick genauer beschrieben werden. Für die Analyse der Ultraschallbilder werden die im folgenden beschriebenen Eigenschaften ausgewertet: Der Kontrast $\sum_{i,j}|i-j|^2C(i,j)$ beschreibt ein Maß für den Intensitätskontrast zwischen einem Pixel und seinem Nachbarn über das komplette Bild gesehen. Die Energie $\sum_{i,j}C(i,j)^2$ ist gegeben durch die Summe der Quadrate der Grauwertmatrix. Die Korrelation $\sum_{i,j}\frac{(i-\mu_i)(j-\mu_j)C(i,j)}{\sigma_i\sigma_j}$ ist ein Maß für die Abhängigkeit der Zeilen und Spalten der Grauwertmatrix untereinander.

Die oben beschriebenen Eigenschaften werden für alle Grauwertmatrizen zu einem Vektor v_{co} zusammengefasst. Um zwei Bildbereiche miteinander zu vergleichen, wird der euklidische Abstand dieser Vektoren berechnet. In Abbildung 6.3 ist die Schätzung des Parameters σ anhand der eben beschriebenen Eigenschaften der Grauwertmatrizen beispielhaft dargestellt.

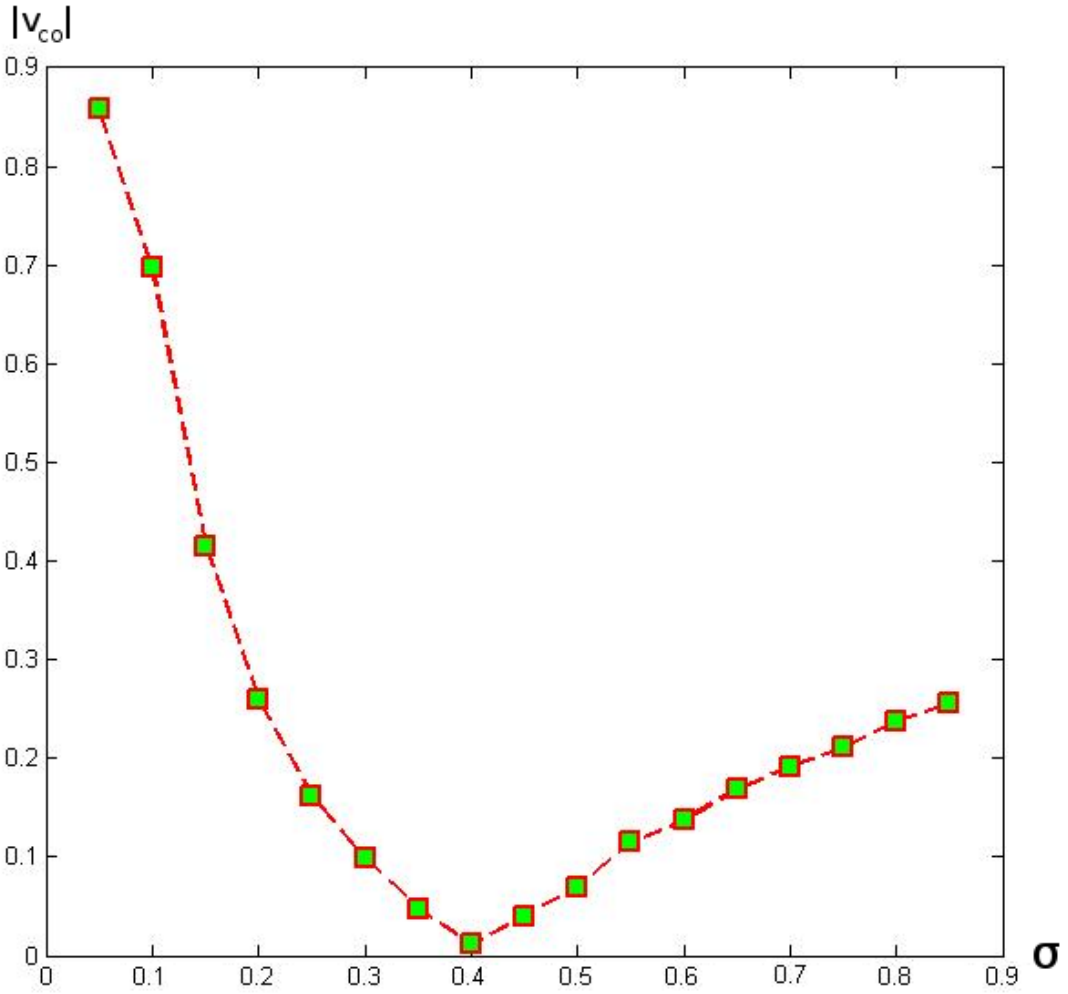


Abbildung 6.3: Beispiel einer Parameterschätzung mit Hilfe von GLCM. In diesem Fall ist das Minimum eindeutig bestimmbar.

Die GLCM wird zur Ermittlung des Streuparameter σ genutzt. Zuerst wird manuell ein frei wählbarer Bildausschnitt selektiert. Anschließend wird dieser Bildausschnitt mit mehreren simulierten Bildausschnitten auf Basis der Summe der Abstände der Vektoren v_{co}^i verglichen, um den Vergleichswert $a = \sum_i |v_{co}^i|$ zu berechnen. Die Variable i gibt hierbei die Ordnungszahl eines simulierten Bildausschnittes an. Durch den folgenden iterativen Prozess wird nun das optimale σ bestimmt. Innerhalb eines Suchfensters, wird eine variabel wählbare Anzahl von Bildern simuliert. Anschließend wird das σ mit dem kleinsten Vergleichswert a als neuer Mittelpunkt des Suchfensters genommen und das Suchfenster wird verkleinert. Dieser iterative Prozess wird solange ausgeführt, bis eine fest vorgegebene Anzahl von Suchläufen überschritten ist oder die Summe der Quadrate der Differenz aller Vergleichswerte einen bestimmten Grenzwert unterschreitet.

Anschließend wird der Streuparameter μ, der maßgeblich die Helligkeit der Speckles bestimmt, ermittelt. Dies geschieht analog zu der Ermittlung von σ, jedoch wird anstatt der Vektoren v_{co} die Differenz der Helligkeit der einzelnen Bildausschnitte als Minimierungskriterium genommen.

Alle zu schätzenden Parameter, Initialwerte und Bildausschnitte können in dem Matlabprogramm über eine graphische Oberfläche 6.4 eingestellt werden.

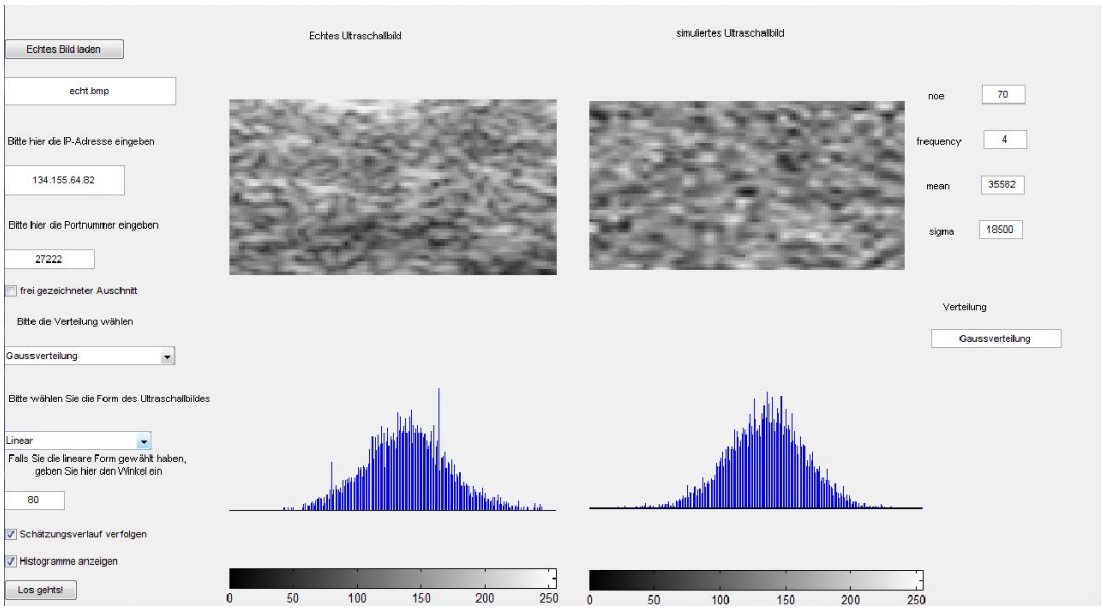


Abbildung 6.4: Screenshot der GUI des Parameterschätzprogramms.

6.4 Szenarienbeschreibung

Ein Szenarium besteht aus der kompletten Beschreibung aller benötigten Objekte und Simulationsparameter. Jedes Szenarium wird durch eine XML-Datei beschrieben, die auf andere Dateien mit unterschiedlichen Formaten verweisen kann. Dieses Format besitzt den Vorteil, dass es leicht von Menschen gelesen und bearbeitet werden kann und sehr gut erweiterbar ist. Demgegenüber steht der Nachteil, dass es einen höheren Speicherbedarf und eine etwas langsamere Verarbeitungsgeschwindigkeit hat, als zum Beispiel ein binäres Format. Auf heutigen Rechnern sind diese Punkte jedoch vernachlässigbar, da die Geschwindigkeit beim Laden des Szenariums nicht relevant ist und der Mehrbedarf an Speicherplatz unerheblich ist. Das XML-Format besteht aus sogenannten Tags die bestimmte Eigenschaften, Werte und weitere Tags enthalten können. Für eine genaue Beschreibung des XML-Formates sei auf (BPSM+06) verwiesen. Im Folgenden ist auszugsweise eine Szenarium-XML-Datei dargestellt:

```
<USS-Scenario>
 <USS-Scene>
  <MEDIUM name="Object1">
   <m_sName>Object1</m_sName>
   <m_eType>Normal</m_eType>
   <m_fSoundSpeed>1473</m_fSoundSpeed>
   <m_fR>1.48</m_fR>
   <m_fAttenuation>1.00dB/cm</m_fAttenuation>
   <m_fMean>0.60</m_fMean>
   <m_fSigma>0.20</m_fSigma>
   <m_fScatterPercentage>0.5</m_fScatterPercentage>
  </MEDIUM>
  [...]
  <OBJECT>
   <name>Object1</name>
   <type>5</type>
   <filename>object.obj</filename>
   <filename_simulation></filename_simulation>
   <mediumName>Object1</mediumName>
   <color_0>0.99</color_0> [...] <color_3>0.5</color_3>
   <scale_0>10</scale_0> [...] <scale_2>10</scale_2>
   <rotation_0>0</rotation_0> [...] <rotation_2>0</rotation_2>
   <translation_0>0</translation_0> [...] <translation_2>0</translation_2>
   <dampingCoefficient>1</dampingCoefficient>
   <volumeStiffness>1</volumeStiffness>
   <stretchingStiffness>1</stretchingStiffness>
   <solverIterations>5</solverIterations>
   <density>1</density>
   <flags>1408</flags>
  </OBJECT>
  [...]
 </USS-Scene>
 <USS-Settings>
  [...]
  <m_cTransducerSettings>
   <m_CbSelectType>1</m_CbSelectType>
   <m_dSbMaxRadius>205</m_dSbMaxRadius>
   <m_sBPixelsPerLineGrid>512</m_sBPixelsPerLineGrid>
   <m_sBNumberOfPhiAnglesGrid>256</m_sBNumberOfPhiAnglesGrid>
   <m_sBPixelsPerLineTransducer>1024</m_sBPixelsPerLineTransducer>
   <m_sBNumberOfPhiAnglesTransducer>256</m_sBNumberOfPhiAnglesTransducer>
   <m_dSbSigmaPulseAxial>1</m_dSbSigmaPulseAxial>
```

```
43    <m_dSbSigmaPulseLateral>5</m_dSbSigmaPulseLateral>
44    <m_dSbSigmaPulseElevation>5.5</m_dSbSigmaPulseElevation>
45    <m_sBSamplePointsAxial>20</m_sBSamplePointsAxial>
46    <m_sBSamplePointsLateral>20</m_sBSamplePointsLateral>
47    <m_sBSamplePointsElevation>10</m_sBSamplePointsElevation>
48    <m_sBSmoothingFactor>0</m_sBSmoothingFactor>
49    <m_dSbStartingRadius>68</m_dSbStartingRadius>
50    <m_sBMinValue>0</m_sBMinValue>
51    <m_sBMaxValue>1</m_sBMaxValue>
52    <m_sBBackgroundValue>0</m_sBBackgroundValue>
53    <m_sBOpeningAngle>39</m_sBOpeningAngle>
54    <m_sBTransducerWidth>40</m_sBTransducerWidth>
55    <m_dSbFrequencyPulse>0</m_dSbFrequencyPulse>
56    <m_pGradientEditor>
57     [...]
58    </m_pGradientEditor>
59    <m_pPulseEditor>
60     [...]
61    </m_pPulseEditor>
62    <m_pPulseScaleEditor>
63     [...]
64    </m_pPulseScaleEditor>
65   </m_cTransducerSettings>
66   <m_cUSSSettings>
67    <m_sBStackSize>14500</m_sBStackSize>
68    <m_sBMaxDepth>20</m_sBMaxDepth>
69    <m_iHashId>316563552</m_iHashId>
70    <m_selectOptiXDevices>0</m_selectOptiXDevices>
71    <m_selectCudaDevices>0</m_selectCudaDevices>
72    <m_dSbSceneEpsilon>1e-12</m_dSbSceneEpsilon>
73    <m_dSbImportanceCutOff>1e-05</m_dSbImportanceCutOff>
74    <m_dSbScatterImportanceCutOff>0.01</m_dSbScatterImportanceCutOff>
75    <m_dSbLambertFactor>1.5</m_dSbLambertFactor>
76    <m_dSbLambertExponent>50</m_dSbLambertExponent>
77    <m_dSbContactGelDiameter>20</m_dSbContactGelDiameter>
78    <m_dSbScaleScatter>0.025</m_dSbScaleScatter>
79    <m_cBAdvanceNoise>0</m_cBAdvanceNoise>
80   </m_cUSSSettings>
81  </USS-Settings>
82 </USS-Scenario>
```

Das erste Tag „USS-Szenario“, enthält alle weiteren Tags und wird als Wurzeltag bezeichnet. Es wird zwingend benötigt, damit der Simulator prüfen kann, ob es

sich um eine gültige Szenarium-Datei handelt. Die zwei erforderlichen Kinder des Wurzeltags sind das „USS-Szene“ und das „USS-Settings“-Tag.

6.4.1 Medium- und Objekteigenschaften der Szene

Die „USS-Scene“ enthält beliebig viele der beiden Untertags „MEDIUM“ und „OBJECT“. Das „MEDIUM“-Tag enthält alle Tags zur Beschreibung eines Mediums und besitzt die optionale Eigenschaft „Name“, welche als Darstellungsname des Mediums verwendet wird. Der Name sollte, muss aber nicht, genauso lauten wie der Wert des Untertags „m_sName“. Die Untertags von „MEDIUM“ sind:

- „m_sName“: ID des Mediums, muss einzigartig sein.
- „m_fSoundSpeed“: Schallgeschwindigkeit in $\frac{\mathrm{m}}{\mathrm{s}}$
- „m_fR“: Schallwiderstand in $\frac{\mathrm{kg} \cdot \mathrm{s}}{\mathrm{m}^2}$
- „m_fAttenuation“: durchschnittliche Abschwächung im Medium in dB/cm
- „m_fMean“: Parameter μ in 4.13
- „m_fSigma“: Parameter σ in 4.13
- „m_fScatterPercentage“: Parameter ν in 4.13

Durch das „OBJECT“-Tag wird ein konkretes Objekt durch die Untertags beschrieben:

- „name“: Eindeutige ID des Objektes.
- „type“: Eigenschaften des Objektes, setzt sich aus Addition der folgenden Werten zusammen: 1-Sichtbar in 3D-Darstellung 2-Aktiv in Ultraschallsimulation 4-Unbewegliches Objekt 8-Deformierbares Physikobjekt 16-Deformierbares Objekt 32-Zylinder-Objekt 64-Kalzifikation 1024-Objekt deaktiviert
- „filename“: Datei mit Geometrieinformationen für die Ultraschallsimulation
- „filename_simulation“: Datei mit Geometrieinformationen für die Physik-Engine
- „mediumName“: ID des verknüpften Mediums
- „color_x“: RGB Farbe des Objektes für 3D Darstellung
- „scale_x“: Skalierung für x,y und z-Achse
- „rotation_x“: Rotation um x,y und z-Achse

- „translation_x“: Verschiebung in x,y und z-Richtung
- „[...]“: verschiedene Einstellungen für die Physik-Engine

6.4.2 Schallkopf-, Simulation- und sonstige Parameter

„USS-Scene“ enthält Tags für allgemeine Einstellungen zur Position des Schallkopfes, der 3D-Ansicht sowie bestimmter GUI-Elemente. Diese sollen nicht manuell gesetzt werden und werden daher im folgenden nicht genauer beschrieben. Auch die Schallkopf- und Simulationsparameter lassen sich am einfachsten über die GUI verändern, werden aber zum besseren Verständnis der Parameter hier kurz aufgelistet. Das Tag „m_cTransducerSettings“ enthält alle Parameter des Schallkopfes:

- „m_CbSelectType“: Typ des Schallkopfes
- „m_dSbMaxRadius“: Maximale Reichweite des Schallimpulses
- „m_sBPixelsPerLineGrid“: Anzahl der Abtastpunkte pro Strahl
- „m_sBNumberOfPhiAnglesGrid“: Anzahl der Strahlen in lateraler Richtung
- „m_sBPixelsPerLineTransducer“: Anzahl Pixel für das simulierte Bild (y-Richtung)
- „m_sBNumberOfPhiAnglesTransducer“: Anzahl Pixel für das simulierte Bild (x-Richtung)
- „m_dSbSigmaPulseAxial“: axiale Ausdehnung der Faltung
- „m_dSbSigmaPulseLateral“: laterale Ausdehnung der Faltung
- „m_dSbSigmaPulseElevation“: elevationale Ausdehnung der Faltung
- „m_sBSamplePointsAxial“: Anzahl der axialen Faltungspunkte
- „m_sBSamplePointsLateral“: Anzahl der lateralen Faltungspunkte
- „m_sBSamplePointsElevation“: Anzahl der elevationalen Faltungspunkte
- „m_sBSmoothingFactor“: Ausdehnung des Glättungsfilters
- „m_dSbStartingRadius“: Startradius für alle Ultraschallstrahlen
- „m_sBMinValue“: Minimaler Grauwert
- „m_sBMaxValue“: Maximaler Grauwert
- „m_sBBackgroundValue“: Grauwert des Hintergrundes

- „m_sBTransducerWidth“: Breite des simulierten Schallfeldes (für Linearen Schallkopf)
- „m_sBOpeningAngle“: Öffnungswinkel des Schallkopfes (für alle anderen Schallköpfe)
- „m_dSbFrequencyPulse“: ϕ Term der Faltung
- „m_pGradientEditor“: gespeicherte Editor-Einstellungen für den Grauwertverlauf.
- „m_pPulseEditor“: gespeicherte Editor-Einstellungen für die TGC.
- „m_pPulseScaleEditor“: gespeicherte Editor-Einstellungen für die zeitabhängige laterale Auflösung

Das Tag „m_cUSSSettings“ enthält alle Parameter des Simulationsprogramms:

- „m_sBStackSize“: Größe des reservierten Stacks für OptiX
- „m_sBMaxDepth“: Maximale Anzahl an Schnittpunkten für einen Strahl
- „m_iHashId“: Hash-Id zur Erkennung der Grafikkarte
- „m_selectOptiXDevices“: GPU Auswahl für OptiX-Berechnungen
- „m_selectCudaDevices“: GPU Auswahl für CUDA-Berechnungen
- „m_dSbSceneEpsilon“: Epsilon für alle Schnittpunkte
- „m_dSbImportanceCutOff“: minimale Intensität eines Strahles für Raytracing-Berechnungen
- „m_dSbScatterImportanceCutOff“: minimale Intensität eines Strahles für Streuungsberechnungen
- „m_dSbLambertFactor“: Lambertfaktor n, siehe Gleichung 4.8
- „m_dSbLambertExponent“: Verstärkungsexponent für Reflexionen, siehe Gleichung 4.9
- „m_dSbContactGelDiameter“: Anzahl der Abtastpunkte, die als Kontaktgel simuliert werden.
- „m_dSbScaleScatter“: Skalierungsfaktor für die 3D-Rauschtextur
- „m_cBAdvanceNoise“: Textur für Blut wird in jedem Bild geändert

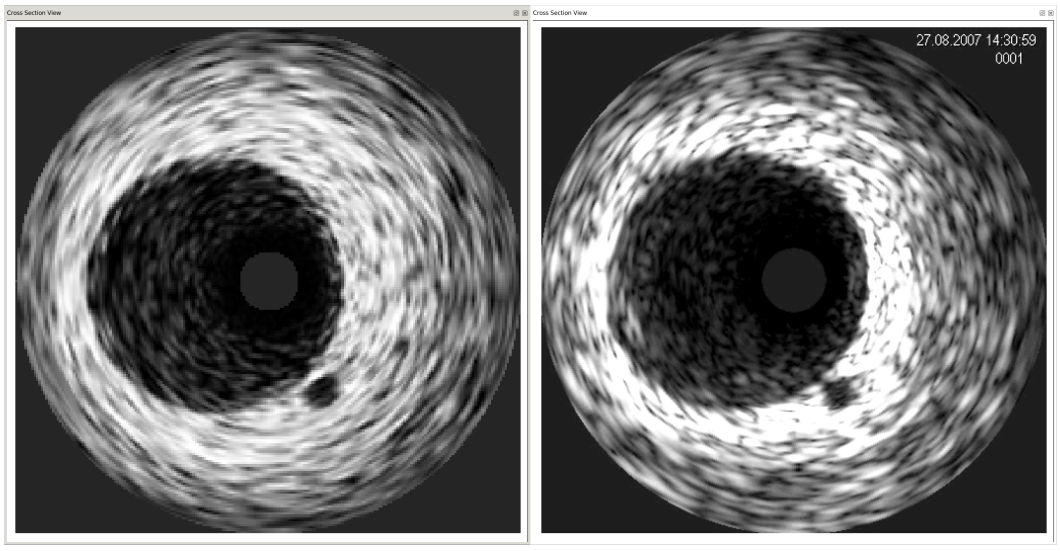
7 Resultate

Für eine Echtzeitsimulation sind zwei Faktoren wichtig, die Realitätsnähe sowie die Performanz. Die Realitätsnähe beschreibt wie realistisch das simulierte Bild im Vergleich zu einem echten Bild aussieht. Die Performanz beschreibt die Geschwindigkeit mit der ein Bild berechnet werden kann.

7.1 2D

7.1.1 Realitätsnähe

Um die simulierten Bilder mit echten Ultraschallaufnahmen vergleichen zu können, wurde mit dem in Abschnitt 6 beschriebenen Editor eine Sequenz von Ultraschallbildern modelliert. Als Ausgangsdaten dienen echte Ultraschallbilder, die mit einem 20MHz Schallkopf aufgenommen wurden. In Abbildung 7.1 ist das simulierte Bild sowie das Original zu sehen.



(a) (b)

Abbildung 7.1: Linke Seite: Ein simuliertes IVUS-Bild eines 20 MHz Schallkopfes. Rechte Seite: Ein echtes IVUS-Bild mit einem 20 MHz Schallkopf aufgenommen.

Es ist schwierig die Ähnlichkeit von zwei IVUS-Bildern mathematisch zu beurteilen, da die Speckle-Struktur nie identisch ist. Daher wird hier nur der Mittelwert und die Varianz der beiden Bilder verglichen, ansonsten werden die beiden Bilder nur visuell verglichen. Der Unterschied des Mittelwerts beträgt weniger als 0.5 Prozent. Der Unterschied der Varianz weniger als zwei Prozent. Die einzelnen Gewebeübergänge sehen realistisch aus. Die Speckle-Struktur ist vorhanden, jedoch sind die einzelnen Speckles im Originalbild besser voneinander zu unterscheiden. Hierdurch wirkt das simulierte Bild insgesamt etwas unschärfer im Vergleich zum Original. Insgesamt ist die Qualität des simulierten Bildes überzeugend und die IVUS Simulation kann in einem Trainingssimulator eingesetzt werden.

Das 2D-Verfahren kann, wie bereits beschrieben, auch für die Simulation von anderen Ultraschallmodalitäten verwendet werden. Dies bietet sich besonders dann an, wenn die Erstellung der exakten 3D-Geometrie schwierig ist und nur Ultraschallbilder von fest definierten Schallkopfpositionen gemacht werden sollen. Im CATHIS®-Simulator wird das 2D-Verfahren zur Zeit genutzt, um transösophageale Echokardiographie zu simulieren. Hierzu wurde eine Sequenz von echten TEE-Aufnahmen segmentiert, die dann im Simulator dargestellt wird. Die Vorteile gegenüber der direkten Verwendung der TEE-Aufnahmen sind folgende:

- die Geometrie kann leicht nach den individuellen Bedürfnissen angepasst werden.
- Instrumente wie Führungsdrähte können in der Ultraschallsimulation berücksichtigt werden.

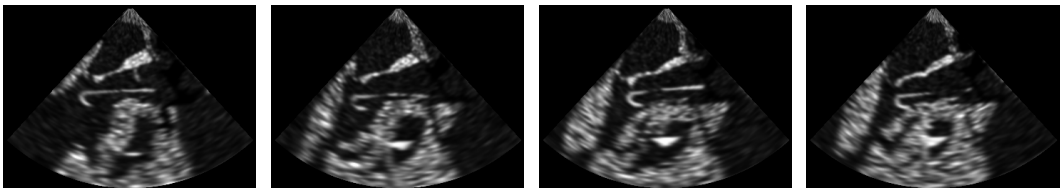


Abbildung 7.2: Sequenz einer 2D-TEE Simulation der Herzkammern mit einem dicken Führungsdraht.

7.1.2 Performanz

Um eine flüssige Bildsequenz zu gewährleisten, muss eine Bildwiederholfrequenz von mindestens 15 Bildern pro Sekunde erreicht werden. Da die Schnittpunktberechnung sehr einfach ist und die erforderliche Faltung nur in 2D ausgeführt wird, benötigt die Neuberechnung eines Bildes nur wenige Millisekunden. Die Echtzeitanforderungen sind hierbei um mehrere Faktoren übertroffen, daher wird an dieser Stelle keine genaue Zeitmessung durchgeführt.

7.2 3D

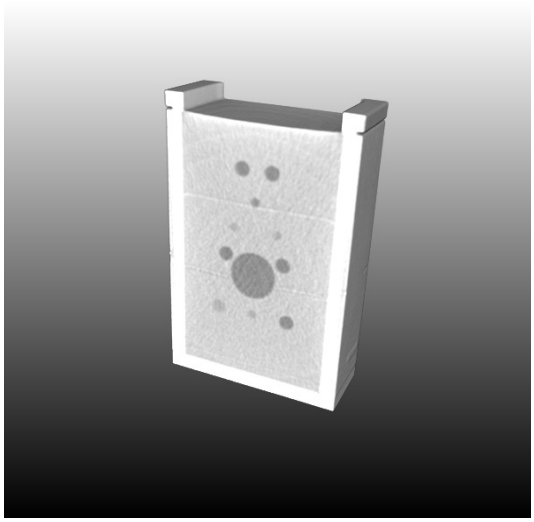
7.2.1 Resultate

Die Simulation wurde in Hinblick auf die Realitätsnähe und in Hinblick auf die Performanz evaluiert. Hierzu wurden verschiedene Szenarien erstellt. Um die Realitätsnähe sowie den Fortschritt im Vergleich zu normalen Schnittbilderverfahren aufzuzeigen, wurden mehrere echte Ultraschallbilder von Phantomen mit simulierten Bildern verglichen. Um die Echtzeitfähigkeit der Simulation, aber auch ihre Grenzen aufzuzeigen, wurden Simulationen mit unterschiedlichen Parametereinstellungen und unterschiedlichem Detailgrad der Objekte in Hinblick auf die Simulationsgeschwindigkeit evaluiert.

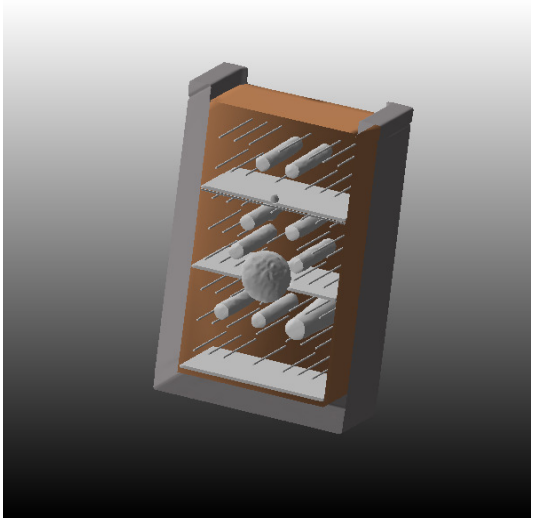
Realitätsnähe

Als erstes Szenarium wurde ein Phantom des Ultraschallsystems BAT®simuliert. Das BAT®(engl. ***B**-mode **A**cquisition and **T**argeting*) ist ein System zur ultraschallbasierten Positionierung und wird in der Strahlentherapie eingesetzt. Das Phantom wird zur Kalibrierung des Positionierungssystems verwendet und besteht aus einem Quader, der mit gewebeähnlichen Materialien gefüllt ist. Innerhalb des Phantoms befinden sich zwei Kugeln und mehrere längliche Kegelstümpfe. Des Weiteren verlaufen mehrere Drähte in fest definierten Abständen durch das Phantom. Von dem Phantom wurde ein CT-Volumen aufgenommen (siehe Abbildung 7.3a). Durch Anwendung der in Kapitel 6.2 beschriebenen Algorithmen wurden aus dem CT-Volumen mehrere Mesh-Objekte erstellt. Jedem der in Abbildung 7.3b dargestellten Mesh-Modelle werden, die in Kapitel 2.2.2 beschriebenen, ultraschallspezifischen Eigenschaften zugewiesen. Da die verwendeten Materialien nicht bekannt sind, wurden diese Werte empirisch ermittelt.

Das Phantom wurde nun mit einem 5 MHz Ultraschallsystem geschallt. Das erzeugte Ultraschallbild ist in Abbildung 7.3c zu sehen. Der virtuelle Schallkopf wurde in die gleiche Lage wie der reale Schallkopf gebracht und ein künstliches Ultraschallbild wurde generiert (siehe Abbildung 7.3d). Die beiden Bilder sind sich sowohl in Form als auch im Aussehen der Strukturen und Speckles sehr ähnlich. Insgesamt wirkt das simulierte Ultraschallbild etwas glatter als das echte Ultraschallbild. Da alle Materialien in dem BAT®-Phantom eine ähnliche Schallgeschwindigkeit und eine ähnliche Schallimpedanz haben, lässt sich kein Unterschied zwischen einem Ray-tracing-Verfahren und einem Verfahren, das Schnittbilder verwendet, erkennen. Lediglich die Drähte bilden hier eine Ausnahme. Diese haben jedoch einen zu kleinen Durchmesser, um signifikante Unterschiede zu erkennen.

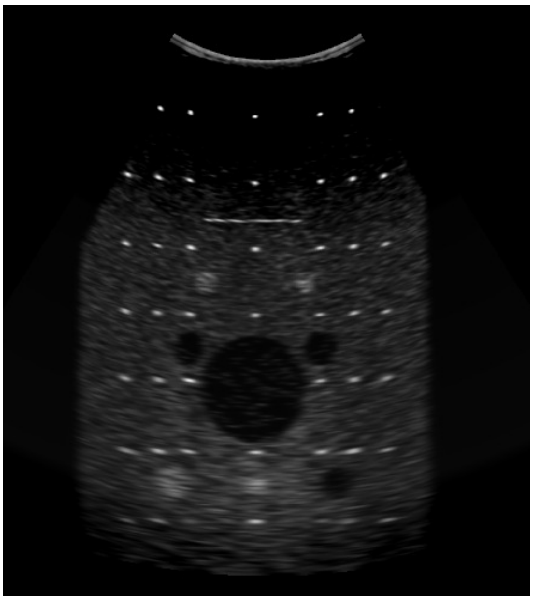


(a) Schnittbild einer CT-Aufnahme des BAT-Phantoms



(b) Schnittbild eines Mesh-Modells des BAT-Phantoms

(c) Ultraschallbild des BAT-Phantoms aufgenommen mit einem konvexen Schallkopf und einer Frequenz von 5 MHz



(d) Simuliertes konvexes Ultraschallbild des BAT-Phantoms mit einer simulierten Frequenz von 5 MHz

Abbildung 7.3: Schnittbilder des BAT-Phantoms und der korrespondierenden Ultraschallbilder. Beide Bilder wurden mit den gleichen Schallkopfpositionen erstellt.

Um den Vorteil des Ray-tracing-Verfahrens, nämlich realistischere Bilder durch Simulation von ultraschallspezifischen Artefakten, gegenüber anderen Verfahren, die Schnittbilder verwenden, zu zeigen, wurde ein eigenes Phantom gebaut. Das

Phantom besteht aus einem rechteckigen Plastikbehälter, der mit Gelatine gefüllt ist. Die Gelatine wurde mit Metamucil angereichert, um ein echoreiches Material zu erhalten (BA95). Innerhalb der Gelatine wurden vier Spritzen nebeneinander platziert. Die Spritzen bestehen aus Polypropylen und sind mit unterschiedlichen Flüssigkeiten gefüllt. Die vier Flüssigkeiten sind Ethanol, Olivenöl, Salzwasser und Wasser. Wasser dient als Referenzmaterial und besitzt ähnliche Ultraschalleigenschaften wie Gelatine. In Ethanol und Olivenöl bewegt sich der Schall mit einer niedrigeren Schallgeschwindigkeit fort, in Salzwasser mit einer höheren Schallgeschwindigkeit. Von jeder Spritze wurden jeweils zwei Ultraschallbilder aufgenommen - eins mit dem Schallkopf orthogonal zur Gelatineoberfläche und eins mit dem einer Schallkopforientierung von 125 Grad (siehe Abbildung 7.4).

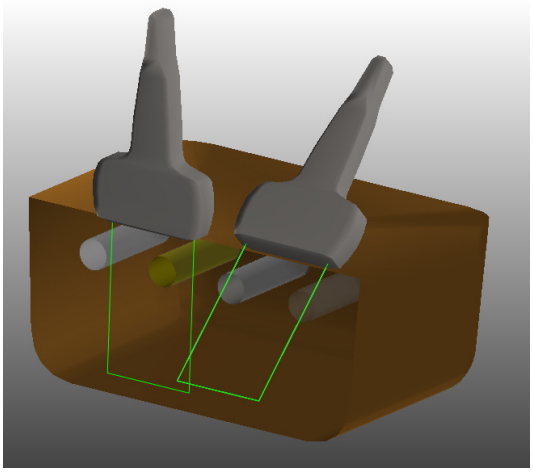


Abbildung 7.4: Schnittbild des Gelatine-Phantoms mit den vier gefüllten Spritzen. Es sind zwei Schallkopfpositionen zu sehen. Eine mit 90° und eine mit 125°.

Nun wurde das Szenarium mit dem Computer simuliert. Die Oberflächenmodelle des Computerphantoms wurden analog zum BAT®-Phantom erstellt. Die ultraschallspezifischen Parameter wurden aus Literaturangaben entnommen (Stö07). Anschließend wurden die künstlichen Ultraschallbilder mit den gleichen Schallkopflagen wie die echten Ultraschallbilder simuliert. Es wurden zwei verschiedene Simulationsverfahren verwendet ein Schnittbild basiertes Verfahren[1] und ein Ray-tracing basiertes Verfahren.

Die jeweils entstanden Bilder (siehe Abbildung 7.6 wurden unter den in Abbildung 7.5 zu sehenden Kriterien ausgewertet:

[1]Die Implementierung basiert auf einer modifizierten Version der Ray-tracing-Methode. Reflexionen und Beugungen des Strahles wurden deaktiviert und die Schallgeschwindigkeit wurde als konstant angenommen.

- δ: Abweichung des Spritzendurchmessers in Ausbreitungsrichtung mit einer orthogonalen Schallkopfstellung
- A_0: Abweichung der Fläche des Schallschattens mit einer orthogonalen Schallkopfstellung
- A_1: Abweichung der Fläche des Schallschattens mit einer Schallkopfstellung von 125°
- Beurteilung der Qualität von Mehrfachreflexionen an der Spritze, benachbarten Spritzen oder des Schallkopfes mit einer orthogonalen Schallkopfstellung

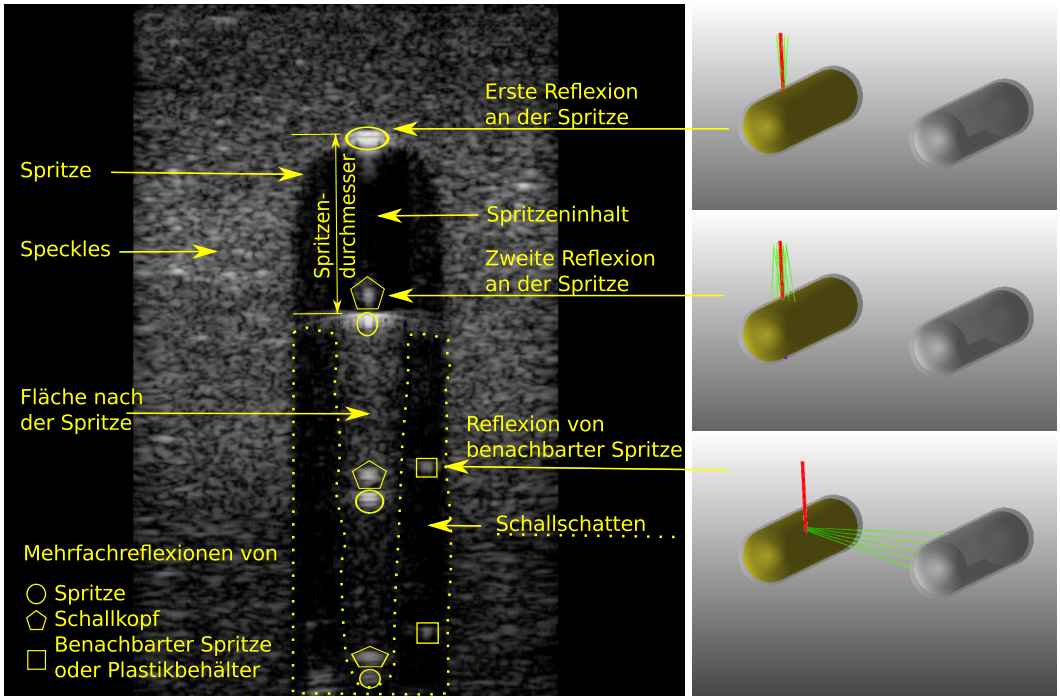
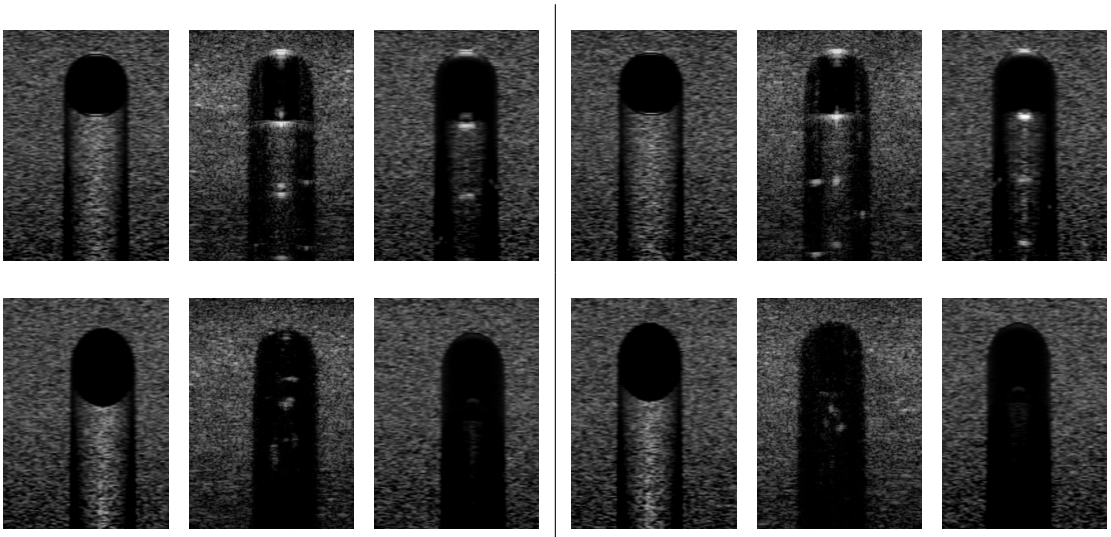
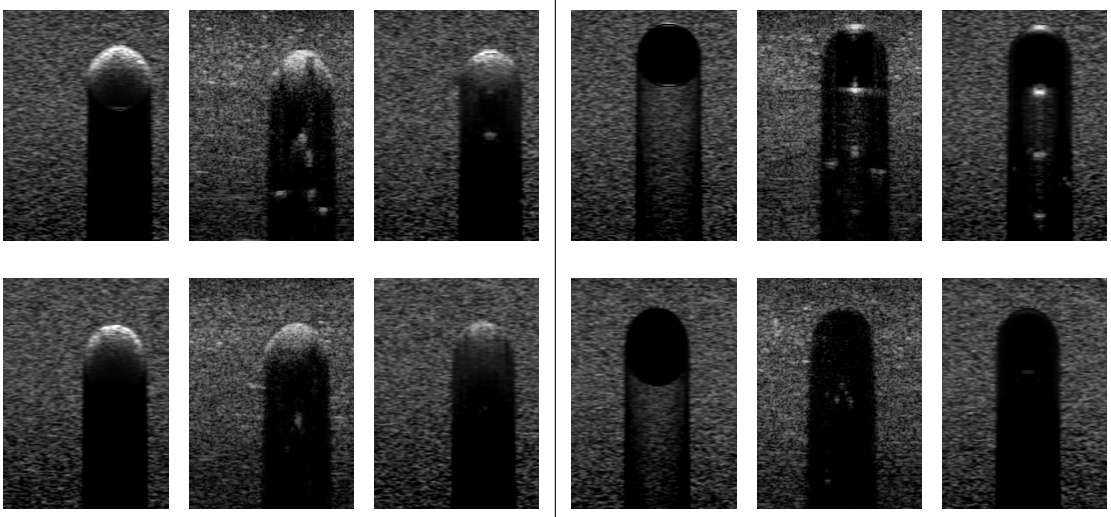


Abbildung 7.5: Beschreibung der Evaluationsparameter des Ultraschallbildes. Auf der rechten Seite kann die Entstehung der ultraschallspezifischen Artefakte gesehen werden.

Die Bilder, die durch die Schnittbild-Methode generiert wurden, zeigen deutliche Abweichungen im Vergleich zu den echten Ultraschallbildern. Der Spritzendurchmesser weicht zum Teil erheblich voneinander ab. Besonders große Unterschiede kann man bei der gemessenen Fläche der Schallschatten feststellen. Auch die Reflexionen bzw. Mehrfachreflexionen fehlen komplett in den Bildern. Im Gegensatz dazu zeigen die mit der Ray-tracing-Methode generierten Bilder nur wenig Abweichungen zu den echten Ultraschallbildern. Der Spritzendurchmesser passt perfekt, es konnten keine Unterschiede zu den echten Ultraschallbildern festgestellt



(a) Schnittbild-Methode (b) Echtes Ultraschallbild (c) Ray-tracing-Methode (d) Schnittbild-Methode (e) Echtes Ultraschallbild (f) Ray-tracing-Methode



(g) Schnittbild-Methode (h) Echtes Ultraschallbild (i) Ray-tracing-Methode (j) Schnittbild-Methode (k) Echtes Ultraschallbild (l) Ray-tracing-Methode

Abbildung 7.6: Vergleich zwischen einer auf Schnittbildern basierten Methode, echten Ultraschallbildern und der Ray-tracing-Methode. Alle Bilder zeigen ein Gelatine-Phantom mit einer gefüllten Spritze. Die Spritzen in (a)-(c) enthalten Ethanol, in (d)-(f) Olivenöl, in (g)-(i) Salzwasser und in (j)-(l) Wasser. Bei den jeweils oberen Bildern hatte der Schallkopf eine Orientierung von 90°, bei den unteren Bildern 125°.

Medium	Schnittbild-Methode	Ray-tracing-Methode
Ethanol	δ ist 13.45% kleiner. A_0 ist 47.30% kleiner. A_1 ist 59.16% kleiner. Mehrfachreflexionen fehlen komplett.	δ ist gleich. A_0 ist 2.32% größer. A_1 ist 0.36% größer. Mehrfachreflexionen stimmen überein. Reflexionen von benachbarten Spritzen sind viel dunkler.
Olivenöl	δ ist 3.74% kleiner. A_0 ist 47.45% kleiner. A_1 ist 61.11% kleiner. Mehrfachreflexionen fehlen komplett.	δ ist gleich. A_0 ist 6.36% größer. A_1 ist 4.13% größer. Mehrfachreflexionen stimmen überein. Reflexionen von benachbarten Spritzen sind viel dunkler.
Salzwasser	δ nicht messbar. A_0 ist 11.33% größer. A_1 ist 2.73% größer. Reflexionen fehlen komplett.	δ nicht messbar. A_0 ist 6.51% größer. A_1 ist 0.64% kleiner. Reflexionen vom Schallkopf stimmen überein. Reflexionen von benachbarten Spritzen sind nicht sichtbar.
Wasser	δ ist gleich. A_0 ist 37.46% kleiner. A_1 ist 33.32% kleiner. Mehrfachreflexionen fehlen komplett.	δ ist gleich. A_0 ist 3.11% kleiner. A_1 ist 0.87% kleiner. Mehrfachreflexionen stimmen überein. Reflexionen von benachbarten Spritzen sind viel dunkler.

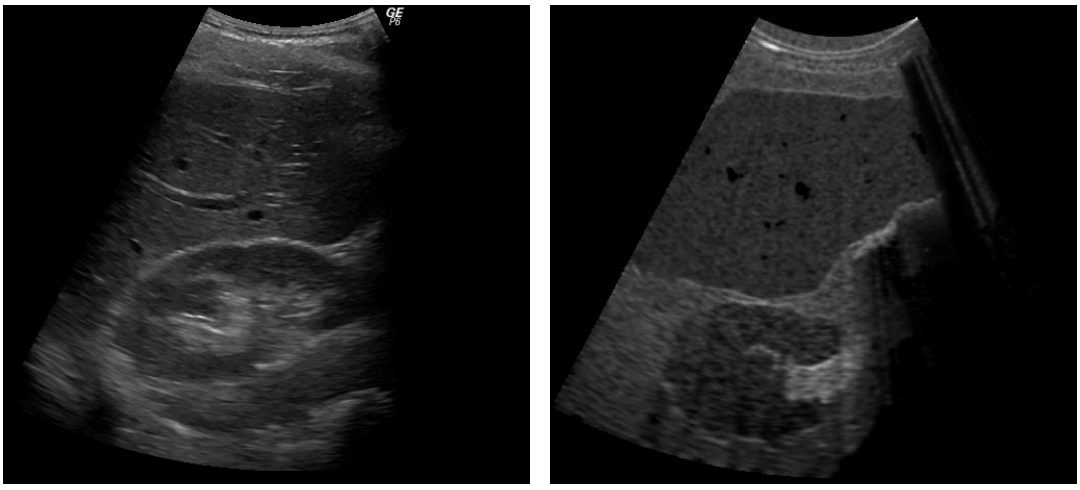
Tabelle 7.1: Unterschiede zwischen echten und simulierten Bildern für Aufnahmen von mit verschiedenen Materialien gefüllten Spritzen.

werden. Die Fläche der Schallschatten unterscheidet sich leicht von den echten Ultraschallbildern. Die größte Differenz ist aber unter sieben Prozent. Die Reflexionen und Mehrfachreflexionen sind meistens an der richtigen Position vorhanden, jedoch deutlich dunkler als in den echten Ultraschallbildern. Insgesamt zeigt sich, dass die Ray-tracing basierte Ultraschallsimulation zwar nicht perfekt ist, jedoch eine deutliche Verbesserung im Vergleich zur Schnittbild basierten Simulation bringt. Eine genaue Auflistung der Unterschiede zwischen simulierten und echten Ultraschallbildern kann aus Tabelle 7.1 entnommen werden.

Um den Nutzen des Simulators zur medizinischen Weiterbildung zu zeigen, wurden zwei weitere Szenarien erstellt. Beim ersten Szenarium handelt es sich um eine Abdomenuntersuchung, beim zweiten Szenarium handelt es sich um eine transrektale Ultraschalluntersuchung der Prostata. Für die Szenarien wird ein Datensatz mit der kompletten Anatomie von Abdomen und Becken benötigt. Die Erzeugung eines solchen Datensatzes ist sehr schwierig und zeitaufwendig und nicht der Schwerpunkt dieser Arbeit. Daher wurde ein geeigneter Mesh-Datensatz gekauft und entsprechend der in Kapitel 6.2 beschriebenen Verfahren

für die Simulation aufbereitet. Ein Nachteil des gekauften Datensatzes ist, dass die Simulationen nur schlecht mit echten Ultraschallbildern verglichen werden können, da die Geometrie der Organe gut übereinstimmen muss, um vergleichbar zu sein.

In der Ultraschallsimulation des Abdomens werden die künstlichen Ultraschallbilder basierend auf der Lage eines virtuellen Schallkopfes in Echtzeit erzeugt. Die Lage des konvexen Schallkopfes kann vom Benutzer über ein haptisches Gerät gesteuert werden. Um das Bild mit dem in Abbildung 7.7a zu sehenden, echten Ultraschallbild zu vergleichen, wurde der virtuelle Schallkopf in die annähernd gleiche Lage gebracht wie bei Aufnahme des echten Ultraschallbildes. Beim Vergleich der beiden Bilder, lassen sich die folgenden Beobachtungen machen:



(a) Sonogramm des Abdomens aufgenommen mit einem 5 MHz Konvexschallkopf

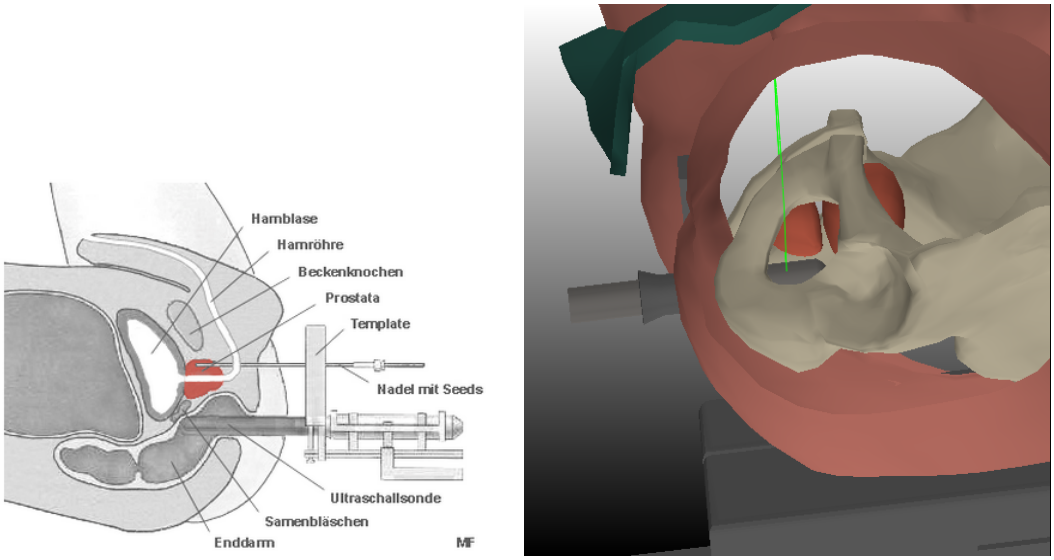
(b) Simuliertes Sonogramm eines virtuellen Patienten.

Abbildung 7.7: Vergleich eines echten Abdomensonogrammes mit einem simulierten Sonogramm.

- Das Speckle-Muster, welches durch kleine Mikroinhomogenitäten erzeugt wurde, sieht realistisch aus.
- Die Reflexionen an den Objektgrenzen sehen realistisch aus.
- Das Aussehen der Leber und der Niere weichen deutlich von den echten Ultraschallbildern ab, da eine Deformationen der Organe durch die Atmung noch nicht implementiert wurde.
- Die Gefäßstruktur sieht in der Simulation nicht so detailliert aus, da die Lebergefäßstruktur des gekauften Datensatzes leider nicht vollständig ist.

Das zweite Szenarium wird zur Simulation einer Brachytherapie der Prostata genutzt. Die Brachytherapie ist ein strahlentherapeutisches Verfahren bei dem

kleinste Strahlungsquellen (engl. *seeds*) aus Jod mit Hilfe von Nadeln in die Zielregion eingesetzt werden.



(a) Skizze einer Prostatabrachytherapie.

(b) Simulation einer Prostatabrachytherapie.

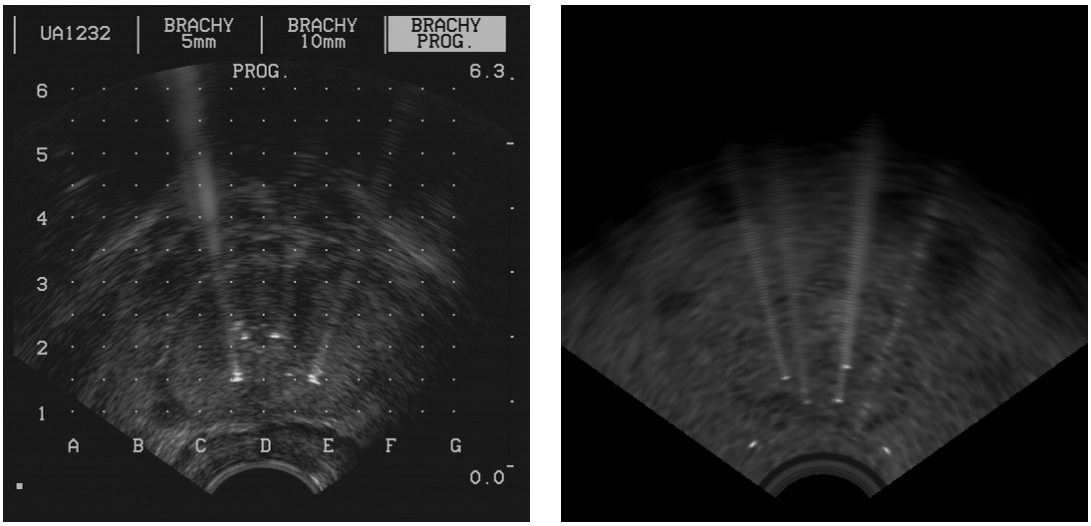
Abbildung 7.8: Aufbau einer typischen Prostatabrachytherapie mit den wichtigsten Strukturen und Geräten.

In Abbildung 7.8 ist der typische Ablauf einer Brachytherapie zu sehen. Der Patient liegt auf einer Liege und ein transrektaler Ultraschallkopf (TRUS) ist an ihm fixiert. Über dem TRUS befindet sich ein Gitter-Template durch das die Nadeln in den Körper gestochen werden. Die Platzierung der *seeds* geschieht unter ständiger Ultraschallkontrolle. Zur Simulation der Nadeln und Deformation des Gewebes können verschiedene Methoden verwendet werden. Die Physik-Schnittstelle ist klar definiert, wodurch eine Physik-Engine mit wenig Aufwand ausgetauscht werden kann. Zur Zeit wird eine Finite-Elemente-Methode (FEM) genutzt, die als Basis die Deformationsbibliothek Vega (SSB13) verwendet.

Die aktuelle Gewebedeformation und die Geometrie der Nadel werden an den Ultraschallsimulator übermittelt, der basierend auf diesen Daten ein transrektales Ultraschallbild generiert. Ein Beispiel eines echten Ultraschallbildes sowie der entsprechenden Simulation sind in Abbildung 7.9 zu sehen. Die Kometenschweifartefakte werden durch die in Kapitel 5.4.2 beschriebenen Modifikationen erzeugt. Es lassen sich die folgenden Beobachtungen machen:

- Das Speckle-Muster sieht im realen Bild etwas feinkörniger aus.
- Die Kometenschweifartefakte sehen auf dem simulierten Bild ein bisschen schärfer aus.

- Das echte Ultraschallbild enthält mehr Strukturen, die im simulierten Bild nicht zu sehen sind.



(a) Beispiel eines Sonogramms während einer Prostatabrachytherapie.

(b) Simuliertes Sonogramm einer Prostatabrachytherapie.

Abbildung 7.9: Vergleich eines echten Sonogramms mit einem simulierten Sonogramm.

Performanz

Die Geschwindigkeit mit der ein Bild simuliert wird hängt nicht nur von den Einstellungen des Schallkopfes, sondern auch von den verwendeten Mesh-Modellen ab. Auch die verwendete Hardware und das genutzte Betriebssystem beeinflussen die Ausführungsgeschwindigkeit der Simulation. Um alle wichtigen Einflussgrößen auf die Performanz zu untersuchen, wurde ein Benchmarkszenarium erstellt. Dieses besteht aus zehn Ellipsoiden, die sich nur durch die Größe und die zugewiesenen Ultraschallparameter unterscheiden (siehe Abbildung 7.10). Alle Simulationsstrahlen durchlaufen alle Ellipsoide. Die Oberflächendiskretisierung jedes Ellipsoides in einem Szenarium besteht aus einer identischen Anzahl von Schichten in Richtung der X- und Y-Achse. Diese Schichten bestimmen die Anzahl der Dreiecke für das Ellipsoid und werden im Folgenden als die Auflösung R des Ellipsoides bezeichnet. Das Szenarium verwendet die im Folgenden verwendeten Parameter, es sei denn, es wird explizit darauf hingewiesen, dass ein Parameter verändert wird.

- Die Schallimpedanz jedes Mediums unterscheidet sich so stark vom Vorgänger, dass eine Reflexion an jedem Schnittpunkt auftritt.
- Bei aktiven Deformationen, werden alle Dreiecke in jedem Zeitschritt um einen zufälligen Wert verschoben.

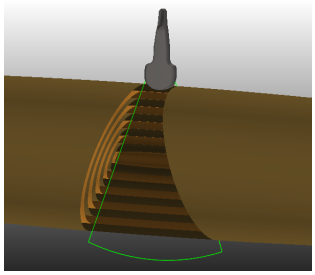
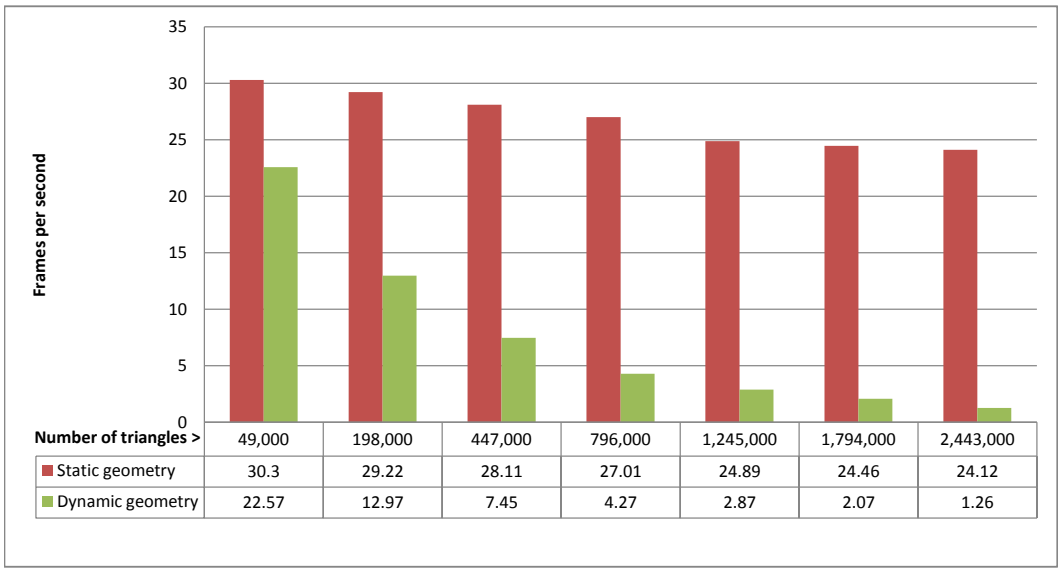


Abbildung 7.10: Schnittbild des Peformanzszenariums mit zehn Ellipsoiden. Es wird ein konvexer Schallkopf simuliert.

- Alle in der Simulation genutzten Materialien haben einen niedrigen Abschwächungskoeffizienten und der Ray-tracing-Prozess wird erst beendet, sobald die Intensität unter 10^{-5} % fällt, um einen zu frühen Strahlabbruch zu verhindert.
- Die Anzahl der Startstrahlen beträgt 5376 und die Anzahl der axialen Abtastpunkt wird auf 512 gesetzt.
- Alle Ellipsoide habe eine Auflösung R von 100 und bestehen aus 19 900 Dreiecken.

Die Parameter der Schallkopfeinstellungen, welche die Simulationsgeschwindigkeit am meisten beeinflussen, sind die Anzahl der Strahlen und die Anzahl der Abtastpunkte pro Strahl. Für diagnostischen Ultraschall wird eine Ultraschallfrequenz von 2 MHz bis 15 MHz verwendet.[2] Für die Simulation solcher Systeme werden 128 bis 512 Abtastpunkte verwendet, um ein Ultraschallbild mit einer passenden Bildtiefe von 1 cm bis 25 cm zu generieren. Die Anzahl der Strahlen pro Bild ist ein Kompromiss zwischen Genauigkeit und Geschwindigkeit. Ein unter Verwendung von wenigen Strahlen simuliertes Bild, kann zwar realistisch aussehen, aber kleinste Strukturen, wie kleine Gefäße, könnten von den Strahlen verfehlt werden. Aus diesem Grund wird die Größe der kleinsten Struktur, welche das Ultraschallbild beeinflusst, berücksichtigt. Diese wird größer als 0,143 mm angenommen. Um die elevationale Ausdehnung einer Schallwelle zu berücksichtigen, wird jeder Strahl mit der Anzahl der Elevationsstrahlen multipliziert. Empirische Versuche haben gezeigt, dass mehr als 256 laterale Strahlen und 21 elevationale Strahlen bei Standardschallköpfen nicht benötigt werden. Dadurch ergibt sich ein Maximum von 5376 Strahlen, die vom Schallkopf aus simuliert werden müssen. Ein anderer wichtiger Performanzfaktor ist der Detailgrad der Mesh-Modelle. Ein detailliertes Gittermodell, mit vielen Dreiecken benötigt mehr Rechenzeit als ein detailarmes Gittermodell mit weniger Dreiecken.

[2] Eine Ausnahme bildet hier intravaskulärer Ultraschall, der mit Frequenzen von bis zu 40MHz arbeitet, jedoch mit einem 2D-US simuliert wird.



Number of triangles >	49,000	198,000	447,000	796,000	1,245,000	1,794,000	2,443,000
Static geometry	30.3	29.22	28.11	27.01	24.89	24.46	24.12
Dynamic geometry	22.57	12.97	7.45	4.27	2.87	2.07	1.26

Abbildung 7.11: Performanzmessungen in Abhängigkeit zur deformierten und undeformierten Geometrie.

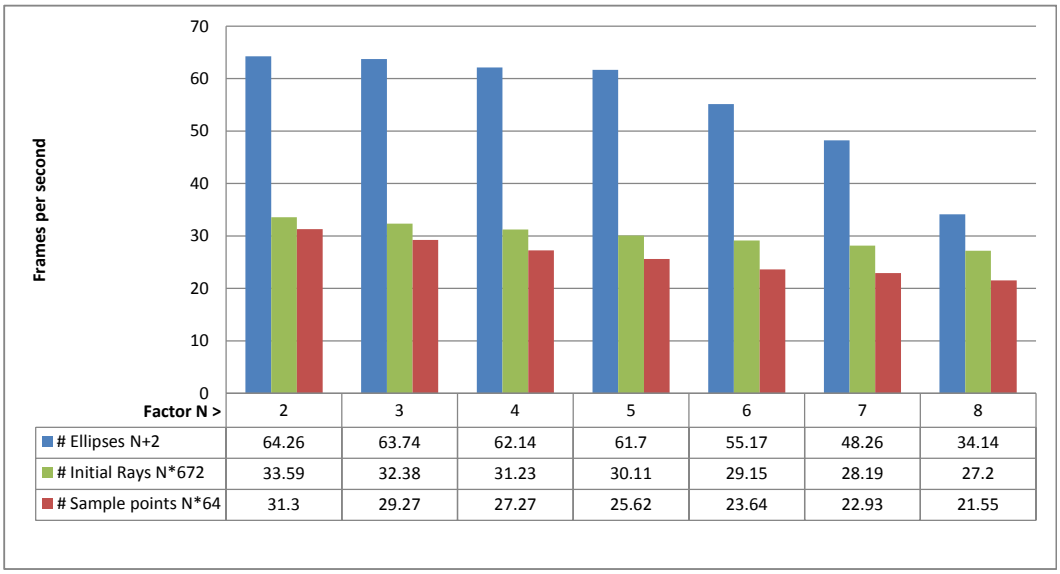
Ein weiterer wichtiger Faktor für die Simulationsgeschwindigkeit ist die Anzahl der Schnittpunkte. Die Schnittpunktanzahl ist ein Parameter, der nicht direkt verändert werden kann, ohne andere Parameter zu beeinflussen, jedoch kann er indirekt verändert werden, indem Ellipsoide deaktiviert werden. Dies geschieht durch eine Veränderung der Position der Ellipsoide. Die deaktivierten Ellipsoide werden so verschoben, dass sie von keinem Strahl mehr geschnitten werden.

Um den Einfluss der aufgezählten Parameter auf die Simulation zu untersuchen, wurden mehrere Messungen durchgeführt, die sich jeweils nur in einem der folgenden Parameter unterscheiden:

- R, und damit die Anzahl der Dreiecke, für statische und dynamische Objekte (Fig. 7.11)
- Die Anzahl der axialen Abtastpunkte, die Anzahl der vom Schallkopf aus simulierten Strahlen und die Anzahl an aktiven Ellipsoiden (Fig. 7.12)

Alle angegebenen Zeitmessungen sind der Durchschnitt von 1000 simulierten Bildern. Die Bilder sind von zufälligen Schallkopfpositionen simuliert, um caching-Effekte zu vermeiden. Alle Messungen wurden auf dem gleichen PC mit der folgenden Hard- und Software verwendet: Intel i7-2600K Prozessor, GTX 590 Grafikkarte[3] Ubuntu 12.04 64-bit und OptiX 2.5.1.

[3] Aufgrund der hohen Synchronisationszeit wurde nur eine der zwei Grafikkarten genutzt.



Factor N >	2	3	4	5	6	7	8
# Ellipses N+2	64.26	63.74	62.14	61.7	55.17	48.26	34.14
# Initial Rays N*672	33.59	32.38	31.23	30.11	29.15	28.19	27.2
# Sample points N*64	31.3	29.27	27.27	25.62	23.64	22.93	21.55

Abbildung 7.12: Performanzmessungen in Abhängigkeit der Anzahl der Strahlen, Abtastpunkte und Ellipsoide.

Ein Nachteil der hier vorgestellten Ray-tracing-Methode im Vergleich zur Schnittbild-Methode ist die längere Ausführungsgeschwindigkeit. Zusätzlich zur Ultraschallsimulation muss noch ein Ray-tracing durchgeführt werden. Eine Untersuchung einer Ultraschallsimulation auf Basis von OptiX wurde bereits von Law et al. (LUK+11) durchgeführt. Die Ergebnisse zeigen, dass die Anzahl der Dreiecke für die Simulation selbst unkritisch ist, sich jedoch sehr stark auf die benötigte Zeit zur Neuberechnung der Beschleunigungsstruktur auswirkt. Die Beschleunigungsstruktur wird zwingend von OptiX benötigt, da die Berechnung der Schnittpunkte ohne eine solche Struktur zu lange dauert. Nähere Details zu den in OptiX verwendeten Beschleunigungsstrukturen finden sich in Kapitel 5.3.1. Dem Paper von Law zufolge wird eine Zeit von über einer Sekunde benötigt, um eine Beschleunigungsstruktur für 49 152 Dreiecke zu aktualisieren, was nicht mit den Echtzeitanforderungen eines Simulators vereinbar ist. In eigenen Messungen hingegen, welche den Standard Beschleunigungsstrukturerzeuger MedianBvH von OptiX verwenden, ist die benötigte Zeit um einige Faktoren niedriger.

Die Performanzmessungen in Abbildung 7.12 zeigen, dass die Anzahl der axialen Abtastpunkte und die Anzahl der zu simulierenden Strahlen die benötigte Simulationszeit linear beeinflussen. Die Anzahl der Schnittpunkte beeinflusst die Simulationsgeschwindigkeit exponentiell. Da jedoch in einem Ultraschallbild selten mehr als 10 Objekte von einem Schallimpuls getroffen werden, ist diese exponentielle Zunahme vernachlässigbar.

Die Anzahl der Dreiecke beeinflusst die Simulation nur leicht, wenn Deformationen deaktiviert sind, jedoch sehr deutlich bei aktiven Deformationen (siehe

Abbildung 7.11). Die Anzahl der Dreiecke ist daher der kritischste Parameter der Simulation in Bezug zur Ausführungsgeschwindigkeit. Bei eingeschalteter Deformation können Modelle mit bis zu 200 000 Dreiecken mit akzeptablen Bildwiederholfrequenzen simuliert werden. Werden nicht alle Objekte verändert und bleibt die Anzahl der veränderten Dreiecke deutlich unter 200 000, so können sogar Modelle mit mehreren Millionen Dreiecken in Echtzeit berechnet werden. Wenn mehr Deformationen benötigt werden, so könnte ein schnellerer Beschleunigungsstrukturersteller implementiert werden. Alternativ könnte man auch die Aktualisierung der Beschleunigungsstruktur für kleine Änderungen abschalten. Die Objekte könnten in mehrere kleine Objekte unterteilt werden, wodurch sich kleine lokale Änderungen nicht auf das gesamte ursprüngliche Objekt auswirken, sondern nur auf wenige kleine Objekte. Bei großen Deformationen, die sich wiederholen, wie zum Beispiel Deformationen aufgrund der Atmung, könnte man auch die Beschleunigungsstruktur vorberechnen und bei Bedarf umschalten.

8 Zusammenfassung und Ausblick

Diese Arbeit beschreibt die Simulation von Ultraschall für medizinische Simulatoren. Es wurden zwei verschiedene Modelle beschrieben, die in Hinblick auf verschiedene Ultraschallmodalitäten entwickelt wurden. Die 2D-Modelle wurden hauptsächlich für intravaskulären Ultraschall entwickelt, können jedoch auch für die Simulation anderer Ultraschallmodalitäten verwendet werden. Die 2D-Simulation bietet sich besonders bei Szenarien an, in denen nur einzelne Schichten von bestimmten Schallkopfpositionen simuliert werden sollen. Der Rechenaufwand sowie die Modellerzeugung sind hier deutlich geringer im Vergleich zur 3D-Simulation. Diese ist jedoch in der Lage Ultraschallbilder aus beliebigen 3D-Positionen zu simulieren.

Der Fortschritt dieser Arbeit gegenüber dem Stand der Technik liegt in der Berücksichtigung der ultraschallspezifischen Artefakte. Durch die Nutzung des 3D-Ray-tracing-Verfahrens ist es erstmals möglich realistische Ultraschallbilder inklusive der typischen Ultraschallartefakte in Echtzeit zu berechnen. Hierdurch wird der Realismus der Simulation enorm gesteigert und es bietet sich an die Simulation vermehrt in der Lehre einzusetzen. Die 3D-Simulation ist auch bei sehr komplexen Szenarien echtzeitfähig.

Die am Anfang der Arbeit gesteckten Ziele für den Ultraschallsimulator wurden alle erfüllt. Im Folgenden werden die Ziele und die Umsetzung dieser Ziele noch einmal aufgeführt und, falls Einschränkungen bestehen, werden diese genannt.

- Speckle-Rauschen: Die simulierten Speckles sind den realen Speckles in Bezug auf Größe, Verteilung und Intensität sehr ähnlich. Die Histogramme der Grauwerte stimmen, wenn auch nicht perfekt, doch sehr gut miteinander überein. Die simulierten Speckles sind insgesamt etwas verschwommener als im Original. Das Aussehen der Speckles verändert sich realistisch, wenn die Lage des Schallkopfes verändert wird.

- Gewebegrenzen: Die Gewebegrenzen der simulierten Bilder sind fast identisch mit denen der echten Ultraschallbilder. Da die Schallgeschwindigkeit in der Simulation für jede Struktur einzeln gesetzt werden kann, sind deutliche Lageabweichung zum Original hier nicht mehr feststellbar. In der Realität hat ein bestimmter Gewebetyp jedoch keine homogene Schallgeschwindigkeit, sondern die Schallgeschwindigkeit variiert auch innerhalb dieses Gewebetyps. Die Inhomogenitäten sind jedoch sehr klein und auch von Mensch zu

Mensch verschieden. Daher können diese kleinen Schwankungen vernachlässigt werden. Auch die Sichtbarkeit von Gewebegrenzen in Abhängigkeit des Schalleinfalles wurde berücksichtigt. Zwar sind auch hier die Intensitäten nicht identisch, da die Berechnung der Reflexionen nur eine Approximation darstellt, aber genau genug für einen medizinischen Simulator.

- Ultraschallartefakte: Durch die Nutzung des 3D-Ray-tracing-Verfahrens können realistische Ultraschallartefakte zu simulieren werden. Andere Algorithmen vernachlässigen diese Artefakte komplett oder können nur vereinzelte Artefakte wie Schallschatten und Schallverstärkung simulieren. Durch das Raytracing-Verfahren ist es möglich, die übrigen Artefakte wie Schichtdicken-, Spiegel-, Mehrfachecho-, Kometenschweif- und Laufzeitartefakte zu simulieren. Die Artefakte stimmen sehr gut mit den in der Realität beobachteten Artefakten überein.

- Echtzeitfähigkeit: Der Anwender kann auch bei komplexen Szenarien nicht erkennen, dass es sich bei den einzelnen Bildern um eine Bildsequenz handelt. Eine Einschränkung, die gemacht werden muss, ist jedoch die Anzahl der Objekte, die gleichzeitig deformiert werden. Beträgt die Summer der Dreiecke von allen deformierten Objekten mehr als 200 000 Dreiecke, dann dauert die Neuberechnung der erforderlichen Beschleunigungsstruktur mehr als 50 ms. Da die Ultraschallsimulation bei solch komplexen Szenarien auch über 30 ms dauert, ist eine Echtzeitberechnung nur eingeschränkt möglich. In Grenzfällen, in denen diese hohe Anzahl an Dreiecken erforderlich ist, kann versucht werden die Objekte in kleinere Objekte zu zerlegen, um undeformierte Regionen nicht unnötigerweise neu berechnen zu müssen.

- Modell der Geometrie: Mit 3D-Oberflächenmodellen können beliebige in der Natur vorkommende Geometrien dargestellt werden. Die Oberflächenmodelle müssen geschlossen sein und es dürfen keine Selbstüberschneidungen vorliegen. Auch eine Überschneidung von verschiedenen Objekten sollte vermieden werden, auch wenn es prinzipiell möglich ist, solche Objekte zu simulieren. Da es sich um ein Standardformat handelt, können die Modelle mit gängigen 3D-Modellierungsprogrammen erstellt, bearbeitet und deformiert werden.

- Ultraschallsysteme: Mit dem Simulator können intravaskulärer, transösophagealer, transrektaler und abdominaler Ultraschall simuliert werden. Andere Ultraschallmodalitäten wie Ultraschall-CT wurden nicht getestet, können jedoch prinzipiell ohne großen Mehraufwand implementiert werden. Der Anwender kann alle Eigenschaften des Schallkopfes zur Laufzeit verändern.

- Bilddarstellung: Das Ultraschallbild wird auf einem Monitor in beliebiger Größe angezeigt. Alle Bildnachbearbeitungsschritte können vom Anwender zur Laufzeit angepasst werden.

- Schallkopfsteuerung: Der Schallkopf ist vom Anwender über verschiedene Methoden steuerbar. Der Schallkopf beim IVUS kann über eine spezielle Hardware der Firma CATHI gesteuert werden. Hierdurch ist eine Handhabung wie in der Realität möglich. Für die Steuerung des transrektalen Ultraschallsystems soll in Zukunft ein echtes Ultraschallsystem der Firma Elekta genutzt werden. Die erforderlichen Anpassungen an die Software wurden bereits gemacht, konnten jedoch bisher noch nicht getestet werden, da nur ein klinisches System zur Verfügung stand, an dem keine Änderungen gemacht werden dürfen. Bei simulierten Abdomenuntersuchungen wird der Ultraschallkopf über ein haptisches Gerät der Firma Sensable gesteuert. Hierdurch kann der Schallkopf fast völlig frei im Raum bewegt werden und Kräfte wie Rückstellkräfte und Reibung können realistisch an den Anwender vermittelt werden. Die Steuerung aller Ultraschallköpfe ist natürlich auch über die GUI möglich.

Der Simulator ist in seinem aktuellen Zustand bereits als Prototyp für das Training von medizinischem Personal einsetzbar. Den Anwendern kann in einem ersten Schritt anhand von konkreten nichtmedizinischen Fallbeispielen das Aussehen und die Entstehung von Ultraschallartefakten beigebracht werden. Der Vorteil gegenüber von Skizzen oder Videos ist, dass der Anwender durch eigenes experimentieren sieht, wie eine Lageveränderung des Schallkopfes das Aussehen eines bestimmten Artefaktes beeinflusst. Durch die Anzeige der simulierten Ultraschallstrahlen mit der größten Energie kann dem Anwender vermittelt werden, auf welchem Weg, in Abhängigkeit zur Zeit, sich der größte Teil der Schallenergie durch das geschallte Medium bewegt. Dies fördert das Verständnis der Herkunft der einzelnen Ultraschallartefakte. Hat der Anwender die grundlegende Funktionsweise der Ultraschallbildgebung verstanden, kann er sein Wissen anhand von konkreten medizinischen Beispielfällen vertiefen. Durch eine 3D-Ansicht auf dem Bildschirm wird dem Anwender die genaue Orientierung des Schallkopfes im Bezug zum durchschallten Körper gezeigt. Durch diese Funktion ist es gut nachvollziehen, wie der Schallkopf bewegen werden muss, um die gewünschte Region zu durchschallen. Die 3D-Ansicht lässt sich durch weitere, bereits implementierte Methoden nutzen, um dem Anwender mehr Informationen über die geschallte Region zu geben. Durch das Aufschneiden der Ansicht entlang der Schallebene ist es dem Anwender möglich, ins virtuelle Innere des Menschen zu blicken. Es können gezielt einzelne Bereiche im simulierten Ultraschallbild, oder in der 3D-Ansicht angeklickt werden, um den Namen der Region zu erfahren. In einem nächsten Schritt könnten weitere Informationen über die selektierte Region angezeigt werden, zum Beispiel welche Artefakte oder auch Anomalien hier häufig auftreten. Die anatomischen Beispielfälle müssen, wie im nächsten Abschnitt beschrieben wird, noch erstellt werden.

Der Simulator kann auch als Lernwerkzeug für andere Anwendungsfälle eingesetzt werden. Konkret wurde mit Hilfe des Ultraschallsimulators bereits ein

Prototyp für einen Brachytherapiesimulator entwickelt. Die Positionierung der Nadel erfolgt, wie in der Realität, über eine echte Nadel. Deren Position wird in Echtzeit durch eine Hardware-Eigenentwicklung der Firma CATHI bestimmt und an die Physik-Engine übermittelt. Diese verarbeitet die Daten und übermittelt der Ultraschallsimulation die neuen Koordinaten aller Objekte. Die neuen Koordinaten werden nun genutzt um eine neues Ultraschallbild zu simulieren und in Echtzeit auf einem Monitor darzustellen.

8.1 Ausblick

Um mit dem Simulator realistische anatomische Fälle simulieren zu können, müssen noch geeignete anatomische Szenarien erstellt werden. Hierfür müssen 3D-Oberflächen erstellt werden, welche die korrekte anatomische Lage der einzelnen anatomischen Strukturen repräsentieren. Eine vollautomatische Simulation und Klassifizierung von bestimmten Regionen würde die Einsatzmöglichkeiten des Simulators deutlich erweitern. In Hinblick auf die Brachytherapiesimulation könnten hier schwierige Eingriffe direkt vor der Operation gezielt geübt werden. Weitere Anwendungsfälle sind auch standardisierte Tests, in denen der aktuelle Wissenstand der Studenten ermittelt wird, um die weiteren Unterrichtsstunden besser daran anpassen zu können. Verschiedene Versuchsreihen können unter den exakt gleichen Bedingungen durchgeführt werden. Oder es können Aufgaben erstellt werden, in denen der Student einen bestimmten Fall untersucht, um anschließend eine Diagnose zu erstellen.

Eine weitere Herausforderung für die Zukunft wird sein, die einzelnen Strukturen korrekt zu deformieren. Insbesondere die Atmung oder das schlagende Herz sind sehr komplexe Bewegungen, die bisher keine Simulation realistisch in Echtzeit simulieren kann.

Im Bereich der Ultraschallsimulation kann die verwendete 3D-Faltung verbessert werden. Eine Möglichkeit, dies zu erreichen, ist die Verwendung von mehr Strahlen. Für jeden der derzeit verwendeten Strahlen können mehrere Strahlen auf einmal berechnet werden, welche die PSF ersetzen. Mit Grafikkarten der aktuellen Generation ist es nicht möglich, diese große Anzahl an Strahlen in Echtzeit zu berechnen. Zusätzliche Erweiterungsmöglichkeiten des Simulators sind jeweils in dem zum Thema passenden Kapitel genannt.

Literaturverzeichnis

[ABHM07] Abkai, Ciamak ; Becherer, Nico ; Hesser, Jürgen ; Manner, Reinhard: Real-time simulator for intravascular ultrasound (IVUS). In: *Proceedings of SPIE* 6513 (2007), Nr. 1, 65131E–65131E–11. http://dx.doi.org/10.1117/12.709115. – DOI 10.1117/12.709115. – ISSN 0277786X

[ACO98] Aiger, D. ; Cohen-Or, D.: Real-time ultrasound imaging simulation. In: *Real-Time Imaging* 4 (1998), August, Nr. 4, 263–274. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.19.8824&rep=rep1&type=pdf

[BA95] Bude, R.O. ; Adler, R.S.: An easily made, low-cost, tissue-like ultrasound phantom material. In: *Journal of Clinical Ultrasound* 23 (1995), Mai, Nr. 4, 271–273. http://dx.doi.org/10.1002/jcu.1870230413. – DOI 10.1002/jcu.1870230413

[BAH07] Bürger, Benny ; Abkai, Ciamak ; Hesser, Jürgen: *Real-Time Simulation of Ultrasound Images based on CT-Volumes*, University of Mannheim, Diss., 2007

[BBRH13] Bürger, B ; Bettinghausen, S ; Rädle, M ; Hesser, J: Real-Time GPU-based Ultrasound Simulation Using Deformable Mesh Models. In: *IEEE transactions on medical imaging* (2013), Dezember. http://www.ncbi.nlm.nih.gov/pubmed/23268382. – ISSN 1558–254X

[BD80] Bamber, J C. ; Dickinson, R J.: Ultrasonic B-scanning: a computer simulation. In: *Physics in Medicine and Biology* 25 (1980), Mai, Nr. 3, 463–79. http://www.ncbi.nlm.nih.gov/pubmed/7403261. – ISSN 0031–9155

[BPSM+06] Bray, Tim ; Paoli, Jean ; Sperberg-McQueen, C. M. ; Maler, Eve ; Yergeau, François ; Cowan, John: *Extensible Markup Language (XML) 1.1 (Second Edition)*. http://www.w3.org/TR/2006/REC-xml11-20060816/. Version: 2006

[Cor11] Corporation, NVIDIA: *NVIDIA CUDA C Programming Guide*. http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:NVIDIA+CUDA+C+Programming+Guide#3. Version: 2011

[Dö8] Dössel, Olaf: *Bildgebende Verfahren in der Medizin*. Springer-Verlag Berlin Heidelberg, 2008

[DEG] DEGUM: *Ultraschallmuseum*. www.ultraschallmuseum.de

[DLT+06] D'Aulignac, D. ; Laugier, C. ; Troccaz, J. ; Vierra, S. ; Vieira, S.: Towards a realistic echographic simulator. In: *Medical Image Analysis* 10 (2006), Nr. 1, 71–81. http://dx.doi.org/10.1016/j.media.2005.02.001. – DOI 10.1016/j.media.2005.02.001

[Ehr98] Ehricke, HH: SONOSim3D: a multimedia system for sonography simulation and education with an extensible case database. In: *European journal of ultrasound* (1998). http://www.sciencedirect.com/science/article/pii/S0929826698000330

[FBC99] Fontaine, I ; Bertrand, M ; Cloutier, G: A system-based approach to modeling the ultrasound signal backscattered by red blood cells. In: *Biophysical Journal* 77 (1999), November, Nr. 5, 2387–2399. http://dx.doi.org/10.1016/S0006-3495(99)77076-1. – DOI 10.1016/S0006–3495(99)77076–1. – ISSN 1542–0086

[FVS11] Fang, Jianbin ; Varbanescu, Ana L. ; Sips, Henk: A Comprehensive Performance Comparison of CUDA and OpenCL. In: *2011 International Conference on Parallel Processing* (2011), September, 216–225. http://dx.doi.org/10.1109/ICPP.2011.45. – DOI 10.1109/ICPP.2011.45. ISBN 978–1–4577–1336–1

[GCC+09] Gao, Hang ; Choi, Hon F. ; Claus, Piet ; Boonen, Steven ; Jaecques, Siegfried ; Van Lenthe, G H. ; Van der Perre, Georges ; Lauriks, Walter ; D'hooge, Jan: A fast convolution-based methodology to simulate 2-D/3-D cardiac ultrasound images. In: *IEEE Trans Ultrason Ferroelect Freq Contr* 56 (2009), Februar, Nr. 2, 404–409. http://dx.doi.org/10.1109/TUFFC.2009.1051. – DOI 10.1109/TUFFC.2009.1051. – ISSN 1525–8955

[GHJV95] Gamma, Erich ; Helm, Richard ; Johnson, Ralph ; Vlissides, John: *Design patterns: elements of reusable object-oriented software*. Addison-Wesley, 1995 (Addison-Wesley Professional Computing Series). http://dx.doi.org/10.1093/carcin/bgs084. http://dx.doi.org/10.1093/carcin/bgs084. – ISBN 0201633612

[GS09] Goksel, Orcun ; Salcudean, S.E.: B-mode ultrasound image simulation in deformable 3-D medium. In: *IEEE Trans Med Imaging* 28 (2009), November, Nr. 11, 1657–1669. http://dx.doi.org/10.1109/TMI.2009.2016561. – DOI 10.1109/TMI.2009.2016561

[GSMS12] Goksel, O ; Sapchuk, K ; Morris, W ; Salcudean, T: Prostate Brachytherapy Training with Simulated Ultrasound and Fluoroscopy Images. In: *IEEE transactions on bio-medical engineering* (2012), Oktober, 1–10. http://dx.doi.org/10.1109/TBME.2012.2222642. – DOI 10.1109/TBME.2012.2222642. – ISSN 1558–2531

[Har79] Haralick, RM: Statistical and structural approaches to texture. In: *Proceedings of the IEEE* 67 (1979), Nr. 5, 786–804. http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1455597

[HG41] Henyey, L.C ; Greenstein, J.L.: No Title. In: *Astrophysical Journal* 93 (1941), S. 70–83

[Hol01] Holm, Sverre: Ultrasim-a toolbox for ultrasound field simulation. In: *Nordic MATLAB conference*, Citeseer, 2001, 5–9

[Jen96] Jensen, J.A.: Field: A program for simulating ultrasound systems. In: *10TH NORDICBALTIC CONFERENCE ON BIOMEDICAL IMAGING, VOL. 4, SUPPLEMENT 1, PART 1: 351–353* Bd. 34, Citeseer, 1996, 351– 353

[Jen97] Jensen, J.a.: A new approach to calculating spatial impulse responses. In: *1997 IEEE Ultrasonics Symposium Proceedings. An International Symposium (Cat. No.97CH36118)* (1997), 1755–1759. http://dx.doi.org/10.1109/ULTSYM.1997.663351. – DOI 10.1109/ULTSYM.1997.663351. ISBN 0–7803–4153–8

[Jen00] Jensen, J: Ultrasound imaging and its modeling. Version: 2000. http://www.springerlink.com/index/NXQETM423MKR620Q.pdf. In: *Imaging of Complex Media with Acoustic and Seismic*. Springer, 2000, 1–38

[JS92] Jensen, J a. ; Svendsen, N B.: Calculation of pressure fields from arbitrarily shaped, apodized, and excited ultrasound transducers. In: *IEEE transactions on ultrasonics, ferroelectrics, and frequency control* 39 (1992), Januar, Nr. 2, 262–7. http://dx.doi.org/10.1109/58.139123. – DOI 10.1109/58.139123. – ISSN 0885–3010

[KDH10] Karimi, K ; Dickson, NG ; Hamze, Firas: A performance comparison of CUDA and OpenCL. In: *CoRR* (2010). http://arxiv.org/abs/1005.2581

[KPY10] Kutarnia, Jason F. ; Pedersen, Peder C. ; Yuan, Christina: Virtual reality training system for diagnostic ultrasound. In: *2010 IEEE International Ultrasonics Symposium* Department of Elec. and Computer Eng., Worcester Polytechnic institute, Worcester, MA 01609, IEEE, 2010. – ISBN 9781457703805, S. 1652–1656

[KS03] Kaufman, D.J. ; Siffert, R.S.: Ultrasound simulation for 3D-axisymmetric models. In: *IEEE Symposium on Ultrasonics, 2003* 00 (2003), Nr. c, 2065–2068. http://dx.doi.org/10.1109/ULTSYM.2003.1293325. – DOI 10.1109/ULTSYM.2003.1293325. ISBN 0–7803–7922–5

[Kut88] Kuttruff, Heinrich: *Physik und Technik des Ultraschalls*. S. Hirzel Verlag Stuttgart, 1988

[KWN10] Karamalis, Athanasios ; Wein, Wolfgang ; Navab, Nassir: Fast ultrasound image simulation using the Westervelt equation. In: *Medical Image Computing and Computer-Assisted Intervention* 13 (2010), Januar, 243–250. http://www.ncbi.nlm.nih.gov/pubmed/20879237

[LPD05] Laguitton, S ; Patard, J.-J. ; Dillenseger, J.-L.: FAST SIMULATION OF ULTRASOUND IMAGES FROM A CT VOLUME. In: *3rd European Medical and Biological Engineering Conference EMBEC* 39 (2005), Nr. 2. http://www.sciencedirect.com/science/article/pii/S0010482508001790

[LUK+11] Law, Y.C. ; Ullrich, Sebastian ; Knott, Thomas ; Kuhlen, Torsten ; Weg, S.: Ultrasound Image Simulation with GPU-based Ray Tracing. In: *Virtuelle und Erweiterte Realität*, 2011, 183–194

[MB95] Meunier, J ; Bertrand, M: Ultrasonic texture motion analysis: theory and simulation. In: *IEEE Trans. Med. Imag.* 14 (1995), Januar, Nr. 2, 293–300. http://dx.doi.org/10.1109/42.387711. – DOI 10.1109/42.387711. – ISSN 0278–0062

[Mor95] Morneberg, Heinz: *Bildgebende Systeme für die medizinische Diagnostik*. Erlangen: Siemens Publicis MCD Verlag, 1995

[MV09] Marion, Adrien ; Vray, Didier: Toward a real-time simulation of ultrasound image sequences based on a 3-D set of moving scatterers. In: *IEEE Trans Ultrason Ferroelect Freq Contr* 56 (2009), Oktober, Nr. 10, 2167–2179. http://dx.doi.org/10.1109/TUFFC.2009.1299. – DOI 10.1109/TUFFC.2009.1299. – ISSN 1525–8955

[NVI11] NVIDIA Corporation: *CUDA C Best Practices Guide*. http://dx.doi.org/10.1145/1296907.1296909. Version: 2011

[PBD+10] Parker, SG ; Bigler, James ; Dietrich, Andreas ; Friedrich, Heiko ; Hoberock, Jared ; Luebke, David ; McAllister, David ; McGuire, Morgan ; Morley, Keith ; Robinson, Austin ; Stich, Martin: Optix: A general purpose ray tracing engine. In: *ACM Transactions on Graphics* 29 (2010), Juli, Nr. 4, 1. `http://dx.doi.org/10.1145/1778765.1778803`. – DOI 10.1145/1778765.1778803

[PBP02] Paluszny, M ; Boehm, W ; Prautzsch, H: *Bezier and B-spline techniques*. Springer-Verlag Berlin Heidelberg, 2002 `http://www.weibnc.com/wp-content/uploads/brkpdfs/Bezier-and-B-Spline-Techniques-by-Marco-Paluszny.pdf`. – ISBN 978–3642078422

[Qt] *Qt Project*. `http://qt-project.org/`

[Ram05] Ramirez, R.: *A Physics-Based Image Modelling of IVUS as a Geometric and Kinematics System.*, Diss., 2005. `http://ddd.uab.cat/pub/tesis/2005/tdx-0809106-134935/mdrr1de3.pdf`

[RH04] Rienstra, SW ; Hirschberg, A: An Introduction to Acoustics / Eindhoven University of Technology. Version: 2004. `http://exocomm.org/library/accoustics/boek.pdf`. 2004 (March). – Forschungsbericht

[RPAS09] Reichl, Tobias ; Passenger, Josh ; Acosta, Oscar ; Salvado, Olivier: Echtzeit-Ultraschallsimulation auf Grafik-Prozessoren mit CUDA. In: *Bildverarbeitung für die Medizin 2009*, Springer, 2009, 157–161

[Sch05] Schwenk, Karsten: *Ein heuristischer Ansatz zur Generierung künstlicher Ultraschallbilder*, Technische Universität Kaiserslautern, Diplomarbeit, 2005

[SHN08] Shams, Ramtin ; Hartley, Richard ; Navab, Nassir: Real-time simulation of medical ultrasound from CT images. Version: Januar 2008. `http://www.ncbi.nlm.nih.gov/pubmed/18982670`. In: *Medical Image Computing and Computer-Assisted Intervention* Bd. 11. Springer-Verlag Berlin Heidelberg, Januar 2008, 734–741

[SHS00] Song, Mingzhou ; Haralick, Robert M. ; Sheehan, Florence H.: Ultrasound imaging simulation and echocardiographic image synthesis. In: *Proceedings of the International Conference on Image Processing*, 2000. – ISBN 0–7803–6297–7, 420–423

[SM98] Stallkamp, Jan ; Matthias, Wapler: UltraTrainer-A Training System for Medical Ultrasound Examination. In: *Medicine Meets Virtual Reality*, 1998, 298–301

[SM11] Sun, Bo ; McKenzie, F.D.: Real-Time Sonography Simulation for Medical Training. In: *International Journal of Education and Information Technlologies* 5 (2011), Nr. 3, 328–335. http://www.naun.org/journals/educationinformation/20-097.pdf

[SPL02] Stippel, Gjenna ; Philips, Wilfried R. ; Lemahieu, Ignace L.: A new denoising technique for ultrasound images using morphological properties of speckle combined with tissue classifying parameters. In: *SPIE Conference Proceedings* 4687 (2002), April, Nr. 25, 324–333. http://dx.doi.org/10.1117/12.462169. – DOI 10.1117/12.462169

[SSB13] Sin, F S. ; Schroeder, D ; Barbic, J: Vega : Nonlinear FEM Deformable Object Simulator. In: *Computer Graphics Forum* 0 (2013), Nr. 32, S. 38–50

[Stö07] Stöcker, Horst: *Taschenbuch der Physik: Formeln, Tabellen, Übersichten.* 5. Frankfurt am Main : Verlag Harri Deutsch, 2007 http://www.bibsonomy.org/bibtex/2631e8854bc402322adbd0d5761072657/hansgeorgbecker. – ISBN 978–3–8171–1720–8 – 3–8171–1720–5

[THL+00] Troccaz, Jocelyne ; Henry, Delphine ; Laieb, Noureddine ; Champleboux, Guillaume ; Bosson, Jean-Luc ; Pichot, Olivier: Simulators for medical training: application to vascular ultrasound imaging. In: *The Journal of Visualization and Computer Animation* 11 (2000), Februar, Nr. 1, 51–65. http://dx.doi.org/10.1002/(SICI)1099-1778(200002)11:1<51::AID-VIS218>3.0.CO;2-8. – DOI 10.1002/(SICI)1099–1778(200002)11:1<51::AID–VIS218>3.0.CO;2–8. – ISSN 1049–8907

[TJRC12] Treeby, B. E. ; Jaros, J. ; Rendell, A. P. ; Cox, B. T.: Modeling nonlinear ultrasound propagation in heterogeneous media with power law absorption using a k-space pseudospectral method. In: *J. Acoust. Soc. Am.* 131 (2012), Nr. 6, S. 4324–4336

[VJHG08] Vidal, By F P. ; John, N W. ; Healey, A E. ; Gould, D A.: Simulation of ultrasound guided needle puncture using patient specific data with 3D textures and volume haptics. In: *Comp. Anim. Virtual Worlds* 19 (2008), Mai, Nr. 2, S. 111–127. http://dx.doi.org/10.1002/cav.217. – DOI 10.1002/cav.217

[WKC+07] Wein, Wolfgang ; Khamene, Ali ; Clevert, Dirk-André ; Kutter, Oliver ; Navab, Nassir: Simulation and fully automatic multimodal registration of medical ultrasound. Version: Januar 2007. http://www.ncbi.nlm.nih.gov/pubmed/18051053. In: *Medical Image Computing and Computer-Assisted Intervention*. Springer Berlin Heidelberg, Januar 2007, 136–143

[ZMRK06] Zhu, Yanong ; Magee, Derek ; Ratnalingam, Rish ; Kessel, David: A virtual ultrasound imaging system for the simulation of ultrasound-guided needle insertion procedures. In: *Proceedings of Medical Image Understanding and Analysis*, Citeseer, 2006, 61–65