

Otto-von-Guericke-Universität Magdeburg



Fakultät für Informatik
Institut für Simulation und Graphik

Bachelorarbeit

Fortgeschrittene Visualisierung für die Therapieplanung

Verfasser:

Elke Grabe

Computervisualisik

25.September 2017

Betreuer:

Prof. Dr.-Ing. Bernhard Preim

Institut für Simulation und Grafik

Dr.-Ing. Michael Sühling

Siemens Healthcare GmbH

Grabe, Elke:

Fortgeschrittene Visualisierung für die Therapieplanung

Bachelorarbeit, Otto-von-Guericke-Universität Magdeburg, 2017.

Selbstständigkeitserklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig und nur mit erlaubten Hilfsmitteln angefertigt habe.

Magdeburg, den 25.09.2017

Elke Grabe

Kurzbeschreibung

In der vorliegenden Bachelorarbeit wird eine Visualisierung entwickelt, die einen Volume Renderer um Funktionen erweitert. Diese Visualisierung wird in einen bestehenden Prototypen eingebaut und soll für Therapieplanungen genutzt werden können. Der Volume Renderer heißt Cinematic Renderer und erstellt fotorealistische Darstellungen von tomographisch aufgenommen Bildern.

Die entwickelten Funktionen beinhalten das Rendern von Segmentierungen mit dem Volume Renderer sowie Visualisierungen, die die Segmentierungen oder andere 3D Modelle ständig sichtbar machen. Zusätzlich umfassen die Erweiterungen die Interaktion mit den 3D Modellen, das semantische Labeln von Segmentierungen und das Exportieren von den Segmentierungen und 3D Modellen im STL-Format. Den Schwerpunkt bildet das Umsetzen einer ständigen Sichtbarkeit auf die Segmentierungen oder 3D Modelle sowie das gemeinsame Rendern mit dem Volume Renderer und diesen Objekten. Für die ständige Sichtbarkeit wird eine Ausschneiden des Volumens durch eine Tiefentextur entwickelt, wodurch die Bereiche des Volumens, in denen die darzustellenden Objekte liegen, freigestellt werden. Dies ähnelt einem Röntgenblick auf die Objekte, die sichtbar sein sollen. Die Objekte werden nicht mit dem Cinematic Renderer dargestellt. Damit diese homogen in die Szene eingebunden werden, wird eine Beleuchtung entwickelt, die dem Cinematic Renderer ähnelt. Das Positionieren der Labels ist durch die entwickelten Erweiterungen in 2D und 3D automatisch möglich.

Die erweiterten Funktionen wurden in einen Prototypen eingebaut und durch einen kleinen Nutzertest evaluiert. Die Ergebnisse zeigen, dass die 3D Visualisierungen für die Patientenaufklärung vor Therapien sinnvoll sind. Durch die gewonnenen Erkenntnisse sind die Vorteile vom Cinematic Rendering für die Therapieplanung offengelegt, weshalb es sinnvoll ist, weitere Entwicklungen in diese Richtung durchzuführen.

Abkürzungsverzeichnis

CT	Computertomographie
MRT	Magnetresonanztomographie
RFA	Radiofrequenz-Ablation
CR	Cinematic Rendering
PCA	Principle Component Analysis
VS	Visual Studio Professional 2013
GUI	Graphische Benutzeroberfläche
MDL	MeVisLab Definition Language
MATE	MeVisLab Advanced Texteditor
MPR	Multiplanare Reformation

Inhaltsverzeichnis

Kurzbeschreibung	iii
Abkürzungsverzeichnis	v
1 Einleitung	1
1.1 Motivation	1
1.2 Aufgabenstellung	1
1.3 Aufbau der Arbeit	2
2 Grundlagen	5
2.1 Medizinischer Hintergrund	5
2.1.1 Akquirierung von medizinischen Bildern	5
2.1.2 Anwendungsfälle	7
2.2 Synthetische Objekte	9
2.3 Interaktive Manipulation von synthetischen Objekten	9
2.4 Visualisierung von tomographischen Daten	9
2.4.1 Cinematic Rendering	10
2.5 Annotationen in der Medizin	13
2.6 Illustrative Volumenvisualisierungsmethoden	14
2.7 Beleuchtung	15
2.7.1 Lokale Beleuchtung	15
2.7.2 Globale Beleuchtung	16
2.7.3 Beleuchtung im Cinematic Renderer	16
3 Konzept	19
3.1 Labeling	19

3.2	Smart Visibility kombiniert mit Cinematic Rendering	20
3.2.1	Schnittebene über Hauptkomponentenanalyse platzieren	21
3.2.2	Ausschneiden mit <i>Clipping Textur</i>	23
3.3	Beleuchtung der synthetischen Objekte	34
3.3.1	Simulation mit mehreren direkten Lichtquellen	34
3.3.2	Platzierung der Lichter in 3D	36
4	Implementierung	37
4.1	Hard- und Software	37
4.1.1	Hardware und Systeminformationen	37
4.1.2	Software	37
4.2	Prototyp	38
4.2.1	Grundgerüst	38
4.2.2	Erweiterungen	40
5	Evaluierung	49
5.1	Versuchsaufbau und Durchführung	49
5.1.1	Fragebögen	49
5.1.2	Ablauf	50
5.2	Ergebnisse und Auswertung	52
5.2.1	Ergebnisse der Smart Visibility kombiniert mit Cinematic Rendering	52
5.2.2	Ergebnisse der Interaktion	56
5.2.3	Ergebnisse der Beleuchtung	57
6	Zusammenfassung und Ausblick	59
6.1	Zusammenfassung	59
6.2	Ausblick	61
6.2.1	Weitere Anwendungen	61
6.2.2	Mögliche Erweiterungen	62
A	Fragebögen	I
	Literaturverzeichnis	V

Kapitel 1

Einleitung

In diesem Kapitel wird zuerst eine Motivation zur Einleitung in das Thema gegeben, dann wird die Aufgabenstellung der Arbeit sowie der Aufbau der Arbeit vorgestellt.

1.1 Motivation

Für die Therapieplanung sind medizinische Bilddaten essenziell. Durch diese können wichtige Therapieentscheidungen getroffen werden. Deshalb sind die Visualisierung der akquirierten medizinische Bilddaten ein bedeutender Bereich in der Informatik. Die Bilddaten werden von Radiologen häufig in 2D Schichtdaten analysiert. Diese sind für Menschen, die nicht so vertraut mit diesen sind, wie Mediziner aus anderen Bereichen oder Patienten, schwer zu verstehen. Durch 3D Darstellungen können räumliche Relationen schneller wahrgenommen werden. Ein Rendering Verfahren, welches Cinematic Rendering (CR) genannt wird, kann die medizinischen Bilddaten in 3D darstellen. Der CR wurde von Siemens Healthcare GmbH entwickelt und erstellt fotorealistische 3D Visualisierungen. Aktuell befindet sich der CR in der Testphase am Univeristätsklinikum Erlangen. Der Klinikdirektor der Chirurgie hat folgende Aussage zum CR gemacht:

„Dank der realitätsgetreuen Cinematic-Rendering-Bilder können alle Kollegen den Patientenfall auf einen Blick erfassen“, sagt Grützmann. „Das Ergebnis sind potentiell schnellere, objektivere und vertrauensvolle interdisziplinäre Therapieentscheidungen – und eine effektivere chirurgische Planung.“ [23]

Damit der CR für die Therapieplanung eingesetzt werden kann, sollte es möglich sein die Geräte, wie z.B. eine OP-Nadel, mit dem vom CR gerenderten Volumen zusammen darzustellen.

1.2 Aufgabenstellung

Ziel

Ziel dieser Arbeit ist es, eine fortgeschrittene Visualisierung für Therapieplanungen mit der neuen Volumenrendering Technik CR zu entwickeln. Dazu soll ein schon bestehender Prototyp um Funktionen erweitert werden.

Funktionsweise

Durch folgenden Punkte soll das Volumenrendering erweitert werden:

- Kombiniertes Rendern von Segmentierung von anatomischen oder pathologischen Strukturen mit tomographisch aufgenommenen Daten

Für Therapieplanungen sind Segmentierungen von anatomischen oder pathologischen Strukturen essenziell, um Strukturen besser hervorzuheben oder zu klassifizieren. Damit die Lage der Segmentierungen im Körper besser verstanden werden können, ist es vorteilhaft, sie mit den tomographisch aufgenommenen Daten zusammen zu rendern.

- Ständige Sichtbarkeit von synthetischen Objekten und interaktive Manipulation von diesen

Die Sichtbarkeit auf die Objekte, die für die Therapie verwendet werden, soll gewährleistet sein, und sie sollen nicht durch das Volumen verdeckt werden. Es soll eine Visualisierungsmethode verwendet werden, die ähnlich einem Röntgenblick ist und die davor liegenden Bereiche ausblendet. Durch die interaktive Manipulation der synthetischen Objekte kann eine Therapie simuliert werden.

- Semantisches Labeln von Strukturen in der gerenderten Szene

Zusätzlich sollen durch gut platzierte Labels Segmentierungen des Volumens benannt werden, sowie Informationen über diese abgebildet werden.

- Export von synthetischen Objekten in STL-Format für den 3D Druck

Zu guter Letzt ist es sinnvoll, die synthetischen Objekte als STL-Format abzuspeichern und zu drucken, damit sie für weitere Planungen oder für die Therapie auch als reales Objekt zur Verfügung stehen können.

Die Erweiterungen sollen in die aktuelle Version des *3D Printing Prototyp* eingebaut werden.

Abgrenzungen

Die Gestaltung der Graphische Benutzeroberfläche (GUI) und der Icons des Prototyps wird in der vorliegenden Arbeit nicht bearbeitet. Zudem wird einzig das direkte Volumenrendering CR für die Therapieplanung in den Prototyp eingebunden und kein anderer direkter Volumen Renderer verwendet.

1.3 Aufbau der Arbeit

Der Rest dieser Arbeit ist in folgende Kapitel aufgeteilt:

Kapitel 1 gibt eine kurze Motivation über das Thema und die Ziel der Arbeit.

Kapitel 2 beinhaltet die Grundlagen, die zur Bearbeitung und zum Erreichen der Ziele notwendig sind. Im Allgemeinen werden medizinische Hintergründe, Begriffe und

Grundlagen zur Visualisierung von medizinischen Daten dargelegt. Dabei wird auf verwandte Arbeiten für die einzelnen Themen eingegangen.

Kapitel 3 beschreibt die Konzepte die zum Erreichen der Ziele umgesetzt werden können.

Kapitel 4 gibt einen Überblick über die verwendete Soft- und Hardware sowie die Implementierung der Konzepte.

Kapitel 5 präsentiert den Ablauf des durchgeführten Nutzertest und die Ergebnisse. Außerdem werden die Ergebnisse ausgewertet.

Kapitel 6 fasst die Zielerreichung der entwickelten Erweiterungen zusammen und gibt einen Ausblick über weitere mögliche Anwendungen sowie Verbesserungsvorschläge zu den Erweiterungen.

Kapitel 2

Grundlagen

In diesem Kapitel werden Grundlagen, auf die die Arbeit aufbaut, beschrieben. Darunter fällt ein kleiner Einblick, wie die Visualisierungen in der Medizin verwendet werden können, eine Begriffserklärung sowie Grundlagen zur Volumenvisualisierungen und Annotationen. Am Ende des Kapitels werden noch illustrative Volumenvisualisierungsmethoden und Beleuchtungsmodelle vorgestellt.

2.1 Medizinischer Hintergrund

Um eine schnelle Heilung zu erreichen sowie unerwünschte Nebenwirkungen und Beschwerden nach einer Operation zu reduzieren, werden immer mehr Operationen minimal-invasiv durchgeführt. Bei minimal-invasiven Verfahren ist es das Ziel die Operationstraumata [24] zu minimieren [39]. Dafür werden mithilfe eines Trokars kleine Einschnitte gemacht, welche durch Tuben (=Rohre) aufgehalten werden, oder Zugänge durch natürliche Körperöffnungen gewählt¹. Nach Herausziehen des Trokars können dann die für die Operation benötigten Instrumente durch die Tuben eingeschoben werden. Zusätzlich kann auch eine kleine Kamera (Endoskop) durch einen der Tuben eingeführt werden, um die Operation im Inneren des Körpers besser kontrollieren zu können. Damit minimal-invasive Operationen erfolgreich ablaufen, ist eine präzise Planung erforderlich. Es muss ein geeigneter Zugangspunkt sowie ein Pfad zum Zielgebiet des Eingriffs gewählt werden. Für diesen Zweck werden im Vorfeld Aufnahmen der Läsion gemacht. Diese Aufnahmen müssen anschließend repräsentativ visualisiert sowie für mögliche Manipulationen (Ausmessen) bereitgestellt werden. Im Folgenden werden Verfahren zur Akquirierung von medizinischen Bildern und typische Erkrankungen, die durch minimal-invasive Eingriffe therapiert werden können, dargestellt.

2.1.1 Akquirierung von medizinischen Bildern

Es gibt verschiedene Verfahren, um medizinische Bilder zu akquirieren. Die für diese Arbeit verwendeten Bilder stammen aus Aufnahmen mit Computertomographie (CT) und Magnetresonanztomographie (MRT), weshalb auch nur diese Verfahren kurz erläutert werden.

¹Diese Verfahren wird *Natural Orifice Transluminal Surgery* (NOTS) genannt [16].

CT wurde zum ersten Mal 1968 von Godfrey Hounsfield vorgestellt. Es ist ein Schnittbildverfahren, d.h. es produziert einen Stapel zweidimensionaler Schichtbilder (griech. *tome*-Schnitt) des akquirierten volumetrischen Objekts. Beim CT werden durch Röntgenstrahlen Gewebe bestrahlt und ein Absorptionsprofil erstellt, welches die Abschwächung der Röntgenstrahlen durch die durchstrahlten Gewebe beinhaltet. Der Patient liegt bei der CT Untersuchung auf einem Tisch, der in eine kurze Röhre (oder Gantry) gefahren wird. In der Gantry ist auf der einen Seite die Röntgenröhre und auf der gegenüberliegenden Seite der Detektor montiert. Die Gantry lässt sich um den Tisch drehen und die Röntgenröhre strahlt dabei fächerförmige Röntgenstrahlen durch den Patienten auf den Detektor. Durch die Rotation der Gantry können aus verschiedenen Winkeln einer einzelnen Schicht des Körpers Bilder aufgenommen werden. Der Tisch wird währenddessen langsam durch die Röhre gefahren, sodass mehrere transversale Schichtbilder akquiriert werden können. Mithilfe einer gefilterten Rückprojektion kann die Abschwächung der einzelnen Voxel² errechnet werden und als zweidimensionales Schichtbild ausgegeben werden [44]. Vorteile von CT sind, dass hochauflösende, detailgenaue Bilder entstehen können und dass die Gantry relativ offen wirkt, da es sich um einen Ring und nicht wie beim MRT um einen Tunnel handelt. Die Scandauer beträgt in der Regel zwischen zehn und sechzig Sekunden. Nachteile von CT ist die Strahlenbelastung für den Patienten und das durchführende Personal, außerdem können Artefakte durch Bewegung des Patienten, Metallimplantate, Partialvolumeneffekt³, Streustrahlen und Strahlaufhärtung auftreten. Deshalb muss vor dem Scannen abgewägt werden, wie hoch die Auflösung sein muss und wie hoch die Dosis sein darf. CT kommt in der Diagnose sowie in der Therapie zum Einsatz und kann besonders gut genutzt werden, um Knochen, den Kopf und die Lunge darzustellen [55].

Magnetresonanztomographie (MRT) ist wie CT ein bildgebendes Verfahren, das Schichtbilder produziert. Es nutzt einen starken Magneten, welcher ein Magnetfeld erzeugt. Durch diesen Magneten werden die Atomkerne der Wasserstoffatome aus dem menschlichen Körper angeregt und richten sich nach dem statischen Feld aus. Durch kurzzeitiges Anlegen von zusätzlich hochfrequenten Wechselfeldern ändern die Wasserstoffatomkerne ihre Orientierung. Nach Abschalten der Wechselfelder richten sich die Atomkerne wieder nach dem statischen Magnetfeld aus. Dies wird Relaxation genannt. Da die Gewebe im Körper zu unterschiedlichen Prozentsätzen aus Wasser bestehen, können über die Relaxationszeit die verschiedenen Gewebe erkannt werden. Diese Zeit wird als Helligkeit im entstehenden Bild gespeichert [55]. MRT Bildgebung hat den Vorteil, dass Menschen bei einer Aufnahme keiner Röntgenstrahlung ausgesetzt ist. Zusätzlich können Bilder mit beliebiger Schichtausrichtung aufgenommen werden und es können Funktionen im Körper, wie der Sauerstofftransport im Blut, aufgenommen werden. Nachteile sind, dass in der Umgebung des Tomographens keine magnetisch anziehende Objekte verwendet werden dürfen, ein MRT sehr teuer ist und die Scandauer im Verhältnis zur Qualität länger ist als beim CT. Besonders gut dargestellt werden können Weichteilgewebe, wie Nerven oder das Gehirn. Einen Überblick über die Funktionsweise von MRT wird in [43] gegeben.

In Abbildung 2.1 ist ein Schichtbild einer CT Aufnahme von einem Thorax und eine Schichtbild einer MRT Aufnahme eines Kopfes zu sehen.

²Volumen Element

³Partialvolumeneffekt entsteht, wenn ein Objekt nur zum Teil in einer Schicht liegt oder wenn das Objekt kleiner als die Schichtdicke ist [55].

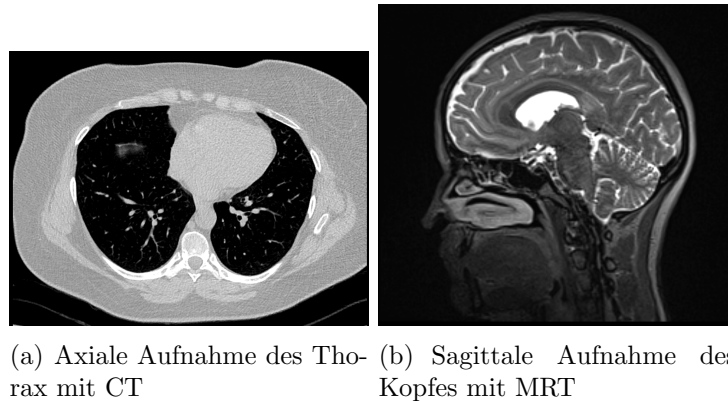


Abbildung 2.1: Vergleich Schichtbilder CT und MRT

2.1.2 Anwendungsfälle

In diesem Abschnitt werden zwei mögliche Anwendungsfälle für die entwickelte Software beschrieben, die einen Nutzen einer verbesserten Visualisierung haben.

Herzklappendefekt

Der Mensch besitzt vier Herzklappen, jeweils eine Segelklappe pro Herzhälfte, die zwischen Vorhof und Kammer liegt und jeweils eine Taschenklappe, die zwischen Kammer und der Aorta oder zwischen Kammer und der Lungenarterie liegt. Die Segelklappen heißen Mitralklappe (links) und Trikuspidalklappe (rechts) und die Taschenklappen heißen Pulmonalklappe und Aortenklappe. Herzklappenfehler lassen sich nach ihren Funktionsstörungen in Stenose⁴ und Insuffizienz⁵ einteilen, oder in eine Kombination aus beiden Funktionsstörungen [6]. Am häufigsten wird eine Stenose der Aortenklappe operiert und diese Erkrankung ist die „dritthäufigste kardiovaskuläre Erkrankung nach der arteriellen Hypertonie und der Koronaren Herzkrankheit“[11]. In Abbildung 2.2 ist der Vergleich einer gesunden und einer erkrankten Aortenklappe zu sehen. Die obere Reihe ist eine gesunde Klappe und die untere eine verengte Klappe. Dabei stellt die linke Spalte die Klappen im geöffneten Zustand und die rechte Spalte im geschlossenen Zustand dar. Es ist klar zu erkennen, dass die erkrankte Klappe nicht mehr so weit geöffnet werden kann wie die gesunde Klappe und, dass im geschlossenen Zustand die erkrankte Klappe nicht mehr komplett schließt. Für die Therapie kann eine mechanische oder eine biologische Herzklappe verwendet werden. Im Verlauf der Arbeit wird die Planung für einen Aortenklappeneratz mit einem 3D Modell einer mechanischen Aortenklappe visualisiert. Das 3D Modell wird von der Firma *On-X Life Technologies*⁶ hergestellt, wurde von D. Herzberg⁷ als rechnerunterstützt konstruiertes (engl. computer-aided design, CAD) Modell nachgebaut und für Bildungszwecke zur Verfügung gestellt. In Abbildung 2.3 ist das Modell zu sehen.

⁴Eine Stenose ist eine Verengung der Herzklappe, wodurch der Blutfluss eingeschränkt ist.

⁵Bei einer Insuffizienz schließt die Herzklappe nicht mehr vollständig, dadurch bleibt mehr Blut im Herzen.

⁶<https://www.onxlti.com/>

⁷<http://www.dan-herzberg.com/>

Abbildung 2.2: Vergleich einer gesunden und einer erkrankten Aortenklappe[47]

(a) (b) (c)

Abbildung 2.3: Verschiedene Ansichten des 3D Modells (Herkunft des Modells[17])

Leberkrebs

„Leberkrebs gehört in Deutschland aktuell zu den wenigen Krebsarten mit steigender Inzidenz und Mortalität.“ [5]

Da die Prognose für die nächsten Jahre nicht besser aussieht, ist es wichtig, die Therapien von Leberkrebs zu verbessern. Dazu zählt auch die Planung von Therapien. Es gibt unterschiedliche Therapien, die bei Leberkrebs zum Einsatz kommen. Welche Therapie durchgeführt wird hängt von unterschiedlichen Kriterien, wie Größe und Anzahl der Tumore, Beschaffenheit der Leber, Alter des Patienten und Zustand des Patienten ab. In dieser Arbeit wird die Radiofrequenz-Ablation (RFA) beschrieben und visualisiert. Bei der RFA wird unter ständiger Bildgebung mittels CT oder Ultraschall eine RFA-Sonde im Zentrum des Tumors platziert. Die Sonde leitet Wechselstrom, der durch einen Radiofrequenzgenerator erzeugt wird und im Tumor Temperaturen bis zu 125°C erzeugt [21]. Durch die entstehende Hitze wird das Tumorgewebe zerstört. Mit dieser Methode können Tumore mit einer Größe von 3-5 cm behandelt werden. Es ist wichtig, dass die Nadel sehr nah am Tumor platziert wird, damit keine gesunden Zellen abgetötet werden. Die Therapie wird unter einer Kurznarkose durchgeführt und die Behandlung erweist sich als gut verträglich [48]. Eine geeignete Visualisierung kann die Planung dieses Eingriffes verbessern und zur Aufklärung der Patienten beitragen.

2.2 Synthetische Objekte

Die vorliegende Bachelorarbeit befasst sich mit der Verbesserung der Sicht auf synthetische Objekte und eine realistischere Beleuchtung von synthetischen Objekten. Bevor auf die Verbesserungsmöglichkeiten eingegangen werden kann, muss klargestellt werden, was mit synthetischen Objekten gemeint ist. Für die Durchführung von Therapien von im Abschnitt 2.1.2 genannten Fällen, gibt es verschiedene Hilfsmittel, die der Arzt benutzen muss bzw. kann. Darunter fallen OP-Nadeln, Katheter und neue Herzklappen. Diese sollen in der Planungsphase einer Therapie durch 3D Modelle visualisiert werden und sind Teil der Objekte, die unter die Kategorie synthetische Objekte fallen. Eine andere Art von Objekten, die dazu zählen, sind Segmentierungen der tomographischen Daten. Ein Beispiel ist die Leber mit ihren Gefäßen und ein darin liegender Tumor. Diese können als *Meshes*⁸ erstellt werden und so in der 3D Szene dargestellt werden.

2.3 Interaktive Manipulation von synthetischen Objekten

Mit der Manipulation von synthetischen Objekten sind Translation, Rotation und Skalierung gemeint. Alle diese Manipulationen sollen möglich sein, damit der Anwender bei einer Therapieplanung keine Einschränkung in der Interaktion mit den synthetischen Objekten hat. Die Interaktion sollte in 2D und 3D durchführbar sein. In 2D sind die Objekte durch die Darstellung der Oberflächenschnitte mit dem Volumen sichtbar. Als Eingabegerät wird eine (2D-)Maus verwendet, welche zwei Freiheitsgrade hat. Dies führt bei der Interaktion mit 3D Objekten zu einem Problem, da diese sechs Freiheitsgrade⁹ besitzen sollen. Damit eine Interaktion aller sechs Freiheitsgrade der Objekte mit einer Maus dennoch möglich sein kann, können *3D Widgets* verwendet werden. Diese sind zusätzliche 3D Objekte, die die Interaktion mit dem eigentlichen Objekt ermöglichen und die Art und Weise, wie mit dem Objekt interagiert werden kann, dem Anwender darstellen [10]. Dafür können zum Beispiel Pfeile in eine Richtung für eine Translation verwendet werden. *Widgets* können auch in 2D die Interaktion vereinfachen. Bei der Rotation zum Beispiel müsste ohne *Widget* vorher ein Rotationszentrum ausgewählt werden und definiert werden, an welchem Punkt das Objekt ausgewählt werden muss, damit dieses rotiert werden kann. Für die Umsetzung der Interaktion in dieser Arbeit wird sich an verfügbaren *Widgets* bedient, welche von der verwendeten Software bereitgestellt werden. Diese werden in Abschnitt 4.2.2 erklärt.

2.4 Visualisierung von tomographischen Daten

Für die Analyse von tomographisch akquirierten Daten sind 3D Volumenvisualisierungen eine gute Erweiterung zu den 2D Darstellungen wie multiplanarer Reformation (MPR),¹⁰.

⁸Ein Mesh ist ein 3D Netz, das aus Knoten, Flächen und Kanten besteht und das die Oberfläche des 3D Modells beschreibt. Häufig wird ein Dreiecks- oder Vierecksnetz verwendet [44].

⁹Translation in x-, y- und z-Richtung und Rotation um diese drei Achsen

¹⁰Ein Rekonstruktionsverfahren um schräge Schnitte durch das akquirierte Volumen zu erhalten. Häufig werden diese sagittal (Blick von der Seite) und coronal (Blick von vorne) gemacht[44]

oder maximum intensity Projektion (MIP)¹¹. Volumenrendering lässt sich in indirektes und direktes Volumenrendering unterteilen. Beim indirektem Volumenrendering (iVR) wird ein Oberflächenmodell vom gegebenen Volumen erstellt. Dafür werden die Voxel des Volumens zum Beispiel nach Farbwerten in zwei Klassen (innerhalb des Volumens und außerhalb des Volumens) eingeteilt. Die extrahierte Oberfläche wird Isofläche genannt und kann mit Hilfe eines Algorithmus in ein Polygonnetz umgewandelt werden. Als Algorithmus lässt sich zum Beispiel der Marching-Cubes-Algorithmus [30] verwenden.

Eine weitere Technik ist das direkte Volumenrendering (dVR). Dies ist ein Verfahren, bei dem die Voxel des Volumens direkt auf Pixel der zweidimensionalen Bildfläche projiziert werden. Für diese Berechnung gibt es verschiedene Algorithmen, die meisten basieren auf dem Prinzip des *Ray Tracings*. Bei diesem wird ein Strahl vom Betrachter ausgesendet und durch die Bildebene und das dahinter liegende Volumen verfolgt. Trifft der Strahl auf einen Voxel, so wird anhand der Materialeigenschaft berechnet, ob der Strahl gebrochen oder reflektiert wird. Wenn der Strahl gebrochen wird, werden neue Strahlen ausgesendet, und wenn der Strahl reflektiert wird, wird der Strahl in eine neue Richtung verfolgt[44]. Die Farbe für den Pixel summiert sich aus den abgetasteten Voxeln. Die Materialeigenschaft wird über eine Transferfunktion bestimmt, welche die Abbildung von den Intensitätswerten auf Farbe oder Transparenz festlegt. Eine Vereinfachung des Algorithmus ist das *Ray Casting*, bei diesem wird die Strahlverfolgung abgebrochen, sobald ein Voxel getroffen wurde. Das *Path Tracing* oder auch *Monte-Carlo-Pathtracing* ist vom gleichen Prinzip, wie das *Ray Tracing*. Es unterscheidet sich darin, dass die Strahlen zufällig generiert werden und nicht pro Pixel ausgesendet werden. Es bildet die Grundlage des im Folgenden erklärten *Cinematic Rendering*. Die Idee des *Path Tracing* stammt von James Kajiya [20].

2.4.1 Cinematic Rendering

Eine neue Art des dVR, das CR, erstellt fotorealistische 3D Darstellungen besser als alle bis dato bekannten Verfahren. Dieses wurde bislang vor allem in der Filmindustrie verwendet, da es die Möglichkeit bietet nachträglich in die Szene eingefügte Charaktere und Gegenstände realitätsnah in die Szenen einzubinden [34]. Einen ersten Ansatz dieses Verfahren für medizinische Daten zu verwenden ist der *Exposure Render*[25]. Siemens Healthcare GmbH nutzt das Cinematic Rendering in kommerziellen Softwareprodukten für die Diagnose von tomographischen Daten. Dabei ist ein Unterschied zur Filmindustrie, dass berechnet wird, wie tief das Licht in Gewebe eindringt und dort in verschiedene Richtungen gestreut wird [13].

Technischer Hintergrund

Das Cinematic Rendering unterscheidet sich zu traditionellen Raytracing Verfahren im Beleuchtungsmodell. Es nutzt physikalisch basierte, globale Beleuchtungsberechnungen. Außerdem bietet es den Vorteil, dass eine frei wählbare Beleuchtungsmap¹² um das Volumen platziert werden kann, und so eine natürliche Beleuchtungssituation entsteht.

¹¹Entlang der virtuellen Kamera werden Strahlen durch das akquirierte Volumen geschickt und der Datenpunkt mit der maximalen Intensität dargestellt.

¹²Eine Beleuchtungsmap ist eine Textur, die eine Umgebung speichert. Diese kann dann auf einen Würfel oder eine Kugel abgebildet werden, um eine 3D Szene zu beleuchten.

In Abbildung 2.4 ist der schematische Ablauf eines verfolgten Strahls dargestellt und wird im Folgenden erläutert. Es ist anzumerken, dass der Ablauf für die Erläuterung stark vereinfacht ist.

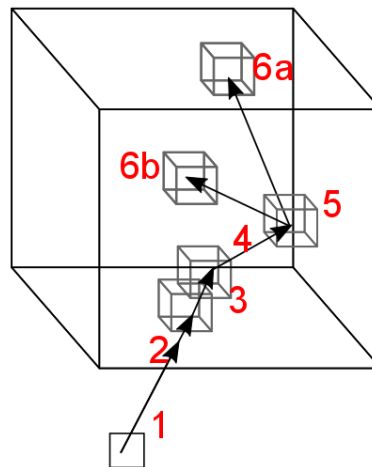


Abbildung 2.4: Schematischer Ablauf einer Strahlverfolgung beim CR

1. Zufällige Wahl eines Pixels und zufällige Generierung eines Strahlurspungs innerhalb des Pixels. Es wird angenommen, dass eine Kamera mit dünner Linse verwendet wird, die die Strahlrichtung bestimmt.
2. Der Eintritt des Strahls in das Volumen wird mit Hilfe des Schnittpunkts zwischen Strahl und Bounding Box ermittelt. Vom Eintrittspunkt wird der Strahl um einen zufällig gewählten Anteil der Schrittweite verschoben.
3. Der Strahl wird dann um eine festgelegte Schrittweite verschoben und das Voxel, das vom Strahl getroffen wird, wird über die Transferfunktion ausgewertet. Der ausgewertete α -Wert des Voxels wird dabei aufsummiert. Der α -Wert eines Voxels beschreibt die Transparenz eines Voxels, je größer er ist, desto undurchsichtiger ist der Voxel. Dann wird der Strahl wieder um die Schrittweite verschoben und der neue α -Werte weiter aufsummiert. Dieser Prozess wird solange wiederholt bis eine zufallsbestimmte obere Schranke erreicht ist.
4. Wenn die obere Schranke der Summe der α -Wert erreicht ist, tritt ein Scatter-Event (deutsch: Streuungsereignis) ein. Das bedeutet, dass der bisher verfolgte Strahl gestreut bzw. reflektiert wird und der Strahl eine neue zufällig generierte Richtung erhält. Die Farbe des zuletzt betrachteten Voxels wird gespeichert und im weiteren für die Bestimmung der Farbe des am Anfang zufällig gewählten Pixels verwendet. Der Strahl wird in die neue Richtung verfolgt und die α -Werte der getroffenen Voxel aufsummiert, wobei die Summe der α -Werte bei jedem Scatter-Event wieder auf null gesetzt wird.
5. Ein neues Scatter-Event tritt ein, wenn die zufallsbestimmte obere Schranke der Summe der α -Werte überschritten wird. Die Farbe des zuletzt betrachteten Voxels wird mit der vorher gespeicherten Farbe multipliziert. Eine neue Strahlrichtung wird generiert.

6. Der neu generierte Strahl wird verfolgt bis:
 - a. die Bounding Box verlassen wird. Dann wird der Lichtfarbwert auf der umgebenden Beleuchtungsmap nachgeschaut, welcher für die Richtung des zuletzt verfolgten Strahls eingetragen ist. Dieser Farbwert wird mit dem bei der Strahlverfolgung errechneten Farbwert multipliziert. Der am Anfang zufällig gewählte Pixel wird nun mit dem Ergebnis der Multiplikation eingefärbt.
 - b. die gegebene maximale Anzahl an Scatter-Events erreicht ist (in diesem Beispiel 2). Damit gilt der Strahl als absorbiert und die resultierende Pixelfarbe ist schwarz.

Der Ablauf einer Pfadverfolgung kann sehr unterschiedlich sein. Durch die Anzahl der erlaubten Scatter-Events kann die Länge des Pfades beeinflusst werden. Je höher die Anzahl der erlaubten Scatter-Events, desto mehr Strahlen werden generiert und der gesamte Pfad wird länger. Ein Pfad kann die minimale Länge eins haben, wenn der erste Strahl nie gestreut wird, also die Summe der α -Werte nie die obere Schranke überschreitet.[34]

Medizinischer Nutzen

Die Visualisierungsart Cinematic Rendering bringt deutliche Vorteile in der Wahrnehmung von 3D Objekten gegenüber üblichen Volumenrendering Methoden. Er beinhaltet viele inhärente Effekte, durch die die räumliche Einordnung von Objekten untereinander verbessert werden kann. Die Effekte sind in der Tabelle 2.1 aufgelistet. Die meisten Volumenrendering Verfahren beinhalten einzelne dieser Effekte, doch das besondere am CR ist, dass er alle diese Effekte inhärent beinhaltet. Das liegt an dem speziellen Path Tracing, dem Monte-Carlo-Pathtracing, wodurch sich die Objekte eines Volumens untereinander beeinflussen und so Schatten im Volumen oder der Farbtransfer entstehen. Die verbesserte Darstellung von räumlichen Zusammenhängen ist für die Therapieplanung ein Vorteil. In Abbildung 2.5 sind zwei der inhärenten Effekte hervorgehoben. In 2.5(a) ist die Tiefenunschärfe außerhalb der Schärfenebene hervorgehoben illustriert und in 2.5(b) ist der Schatten der innerhalb des Volumens geworfen wird durch das rot umrandete Rechteck hervorgehoben. Ein anderer Vorteil ist, dass die Berechnung im CR sofort nach Änderungen der Parameter, wie Beleuchtungsmap, Transformation der Kamera oder des Volumens oder der Transferfunktion, geschieht. Dadurch erhält der Nutzer sofort ein visuelles Feedback, was für den medizinischen Einsatz des CR notwendig ist und gerade für die Therapieplanung einen großen Vorteil bietet [34].

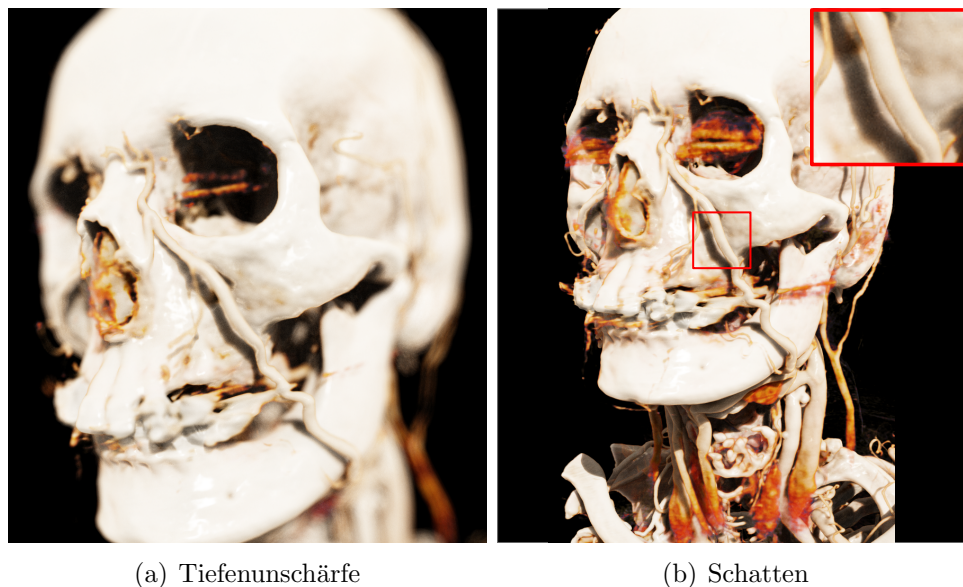


Abbildung 2.5: Schädel Datensatz vom CR gerendert zum Hervorheben der inhärenten Effekte

2.5 Annotationen in der Medizin

Um Vorabinformationen oder bei der Befundung erfahrene Informationen immer sichtbar zu haben, können Annotationen in Form von kurzen Notizen verwendet werden. Die Notizen können in Textform (Labels) dargestellt werden [37]. Anwendungsgebiete sind die Lehre der Anatomie und die Dokumentation von Befundungen. In dieser Arbeit werden Labels verwendet, um die Informationen aus der Befundung für die Therapieplanung bereitzustellen. Dass Labels für die Therapieplanung nützlich sind, wird in [38] diskutiert. Damit die Labels dem Benutzer helfen, muss die Platzierung richtig gewählt werden. Anforderungen an solche Labels und die Umsetzung wurden erstmals in [45] präsentiert. Die Platzierung der Labels ist für 2D und 3D Visualisierungen

Effekt	Bedeutung
Ambiente Occlusion	niedrigere Lichtintensitäten in Ritzen und Ecken durch Selbstbeschattung
Depth of Field	Tiefenunschärfe außerhalb der Schärfenebene der simulierten Kamera
Color Bleeding	Farbtransfer zwischen Oberflächen durch indirekte Beleuchtung
Schatten	Schattenwurf zwischen verschiedenen Objekten

Tabelle 2.1: Inhärente Effekte [34]

wichtig und kann entweder durch den Benutzer interaktiv oder automatisch erfolgen. Es wird zwischen internen Labels, welche direkt auf dem zu beschriftendem Objekt platziert sind, oder externen Labels, welche am äußeren Rand des Bildes platziert sind, unterschieden. Interne Labels haben den Vorteil, dass sie nah am zu beschreibenden Objekt liegen, aber dadurch auch Teile im Bild überlagern. Externe Labels benötigen einen Ankerpunkt¹³ und eine Verbindung zu dem zu beschreibenden Objekt. Die Verbindung kann durch eine (gestrichelte) Linie und/oder durch die Wahl der gleichen Farbe vom Ankerpunkt und der Schriftfarbe des Labels gezogen werden [44]. Da in der Medizin die exakte Analyse der aufgenommenen Bilder wichtig ist und diese durch Überlagerung von internen Labels nicht immer gewährleistet sein kann, werden in dieser Arbeit nur externe Labels verwendet.

2.6 Illustrative Volumenvisualisierungsmethoden

„Illustrative visualization has the main purpose to communicate information and not necessarily to look real.“ [18]

Das Zitat stellt das Hauptziel von illustrativen Visualisierungen dar. Mit illustrativen Visualisierungsmethoden sollen gegebene Daten so repräsentiert werden, dass durch sie weiteres Wissen über das aufgenommene Objekt dem Betrachter kommuniziert wird. Dabei sollen die Visualisierungen nicht zu komplex sein, aber auch keine wichtigen Daten vernachlässigt werden. Im Zusammenhang mit medizinischen Daten können illustrative Visualisierungsmethoden eingesetzt werden, um komplexe Strukturen zu verstehen. Durch diese Methoden soll Wissen über die räumlichen Zusammenhänge von Gefäßen, Organen, Knochen und anderen Strukturen der medizinischen Aufnahmen erlangt werden. Sicherzustellen ist auch, dass wichtige Objekte nicht von anderen Strukturen verdeckt werden. Eine Art der Methoden wird unter *Smart Visibility* zusammengefasst und wird umfangreich in [56] und in [44] dargestellt. Im Folgenden werden einige Methoden vorgestellt, die für die Anwendungsfälle dieser Arbeit von Bedeutung sind.

Eine Methode, um Objekte von hoher Wichtigkeit sichtbar zu machen, wird in [57] beschrieben. Bei dieser Technik bekommt jeder Teil der Daten eine Gewichtung in Abhängigkeit zur Sichtbarkeit des jeweiligen Teils. Anhand der Gewichtung werden dann die Teile mit einem hohen Wert sichtbar dargestellt, auch wenn sie von einem anderem Objekt überlappt werden. Diese Technik wird *Importance-Driven Volume Rendering* genannt.

In [52] werden mehrere Methoden zur Erlangung der Sichtbarkeit auf wichtige Objekte diskutiert. Eine davon ist die *Distanzbasierte Transparenz*, welche von A. Tappenbeck entwickelt wurde und in [51] genauer beschrieben ist. Bei dieser werden die Abstände von wichtigen Objekten zu den angrenzenden Strukturen berechnet und in einem vorher festgelegten Radius ausgeblendet, oder wenn die Strukturen nicht in dem Radius liegen, werden sie in Abhängigkeit zur Distanz weniger transparent dargestellt.

Eine andere Methode, die die Sichtbarkeit auf das zu betrachtende Objekt garantiert,

¹³Damit ist eine Punkt auf dem Bild gemeint, der durch seine Koordinaten charakterisiert wird. Meistens wird der Mittelpunkt des Objektes gewählt.

nennt sich *Cut Away*. Bei dieser Technik werden alle Objekte, die zwischen dem zu betrachtenden Objekt und dem Nutzer liegen, ausgeblendet. Dies passiert nur regional und häufig in Form eines Zylinders oder Quaders. Ein Nachteil des *Cut Away* ist, dass eine harter Übergang zwischen dem ausgeschnittenem Teil und dem restlichem Objekt entsteht, weshalb die räumlichen Zusammenhänge schwer zu deuten sind.

Oft werden *Cut Aways* zusammen mit dem *Ghosting* verwendet. Beim *Ghosting* werden Objekte, die vor dem zu betrachtendem Objekt liegen, nicht komplett ausgeblendet, sondern transparenter dargestellt. Dadurch bleiben die räumlichen Zusammenhänge einfacher zu deuten. Wichtig ist, dass dem Betrachter genügend viele Teile der Objekte sichtbar sind, sodass die Teile der Objekte mental zusammengefügt werden und als ein komplettes Objekt wahrgenommen werden. Dies ist in [8] ausführlich beschrieben.

Eine andere Herangehensweise, um die Sichtbarkeit auf ein Objekt in einem Volumen zu erlangen wird in [7] von Bruckner et al. vorgestellt. Die Methode ist eine Erweiterung zum Volumen Rendering, welche die Opazität verändert. Regionen, die dem Licht zugewandt sind, also stark beleuchtet sind, werden transparenter dargestellt im Vergleich zu Regionen, welche schwächer beleuchtet sind. In Folge dessen erscheinen Oberflächen, die zum Betrachter ausgerichtet sind, durchsichtig und Kanten, die orthogonal zum Betrachter sind, sichtbar. Für die Berechnung wird die Opazität aus der Intensität von Schatten, der Gradientengröße, der Distanz zum Betrachtungspunkt und der im Volume Rendering errechneten Opazität zusammengerechnet. Diese Methode wird *Illustrative Context-Preserving Volume Rendering* genannt.

2.7 Beleuchtung

Beleuchtung spielt in der Volumenvisualisierung eine große Rolle. Durch Beleuchtung und den von den Objekten geworfenen Schatten kann der Mensch die räumliche Relation von Objekten untereinander erkennen. Dieser Vorgang wird „Formwahrnehmung aus Schattierung“ genannt [40]. Außerdem kann der Mensch durch die Beleuchtung und die Reflektion der Objekte Rückschlüsse auf die Beschaffenheit der Oberfläche ziehen. Deswegen ist die Beleuchtung von 3D Objekten für die Therapieplanung in der Medizin entscheidend. Es gibt verschiedene Beleuchtungsmodelle die sich in lokale und globale Beleuchtungsmodelle unterteilen.

2.7.1 Lokale Beleuchtung

Bei der lokalen Beleuchtung wird das Auftreffen von einer Punktlichtquelle auf die Oberfläche von einem Objekt simuliert. Schatten wird dabei nicht erzeugt. Ein einfaches lokales Modell ist das *Lambert-Beleuchtungsmodell* oder auch *diffuse Reflexion* genannt, bei diesem wird das Licht mit gleicher Intensität in alle Richtungen gestreut. Dadurch werden alle Objekte der Szene mit gleicher Intensität beleuchtet [27].

Ein anderes Modell, welches sehr häufig verwendet wird, wird *Phong-Beleuchtungsmodell* genannt. Die Farbe für jeden Pixel eines Bildes wird mit der Gleichung 2.1[44] berechnet:

$$I_{gesamt} = I_{ambient} + c_{att} \sum_{i=0}^{n-1} (I_{diffusion,i} + I_{spekular,i}). \quad (2.1)$$

Die Gleichung setzt sich aus drei Komponenten zusammen. Die erste Komponente ist der Anteil der Intensität der *Ambienten Beleuchtung*. Bei dieser wird angenommen, dass jedes Objekt mit der gleichen Lichtintensität bestrahlt wird, und es eine zweite Komponente gibt, den *ambienten Reflexionskoeffizienten*. Dieser liegt zwischen null und eins und gibt die Materialeigenschaft eines Objektes an [50]. Die Reflexion von diffuser und spekulärer Beleuchtung ist abhängig von der Lichtquelle, deswegen wird über alle Lichtquellen von *null* bis *n* iteriert. Der Faktor c_{att} sorgt dafür, dass Objekte mit derselben Orientierung aber unterschiedlichen Positionen voneinander unterschieden werden können. Die *spekulare Reflexion* beschreibt die Reflexion von glänzenden Flächen. Diese ist im Gegensatz zur diffusen Reflexion nicht an allen Voxeln gleich, sondern abhängig von der Position des Beobachters. Die spiegelnde Stelle wird in der Farbe des Lichtes reflektiert, der Rest des Objektes wird mit diffusem Licht reflektiert. Eine ausführlichere Beschreibung des *Phong-Beleuchtungsmodell* kann in [44] nachgelesen werden.

Da es in der vorliegenden Arbeit darum geht, die synthetischen Objekte möglichst real zu beleuchten und lokale Beleuchtungsmodelle den Nachteil haben, Objekte nicht realistisch zu beleuchten, wird keines der vorgestellten lokalen Beleuchtungsmodelle verwendet.

2.7.2 Globale Beleuchtung

Die globale Beleuchtung unterscheidet sich grundsätzlich zur lokalen Beleuchtung, da bei diesen Modellen nicht nur die Reflexion vom beleuchteten Objekt betrachtet wird, sondern auch die Wechselwirkung der Reflexionen von anderen Objekten der Szene. Ein schon beschriebenes Verfahren ist der *Cinematic Renderer*. Weitere Verfahren sind herkömmliche Volumen Rendering Verfahren, wie *Ray Tracing* oder *Path Tracing*, die schon in 2.4 beschrieben wurden. Ein etwas anderes Verfahren ist die *Radiosität*. Bei diesem wird im Gegensatz zu den vorher genannten Verfahren nicht die Lichtausbreitung durch Lichtstrahlen simuliert, sondern es beruht auf dem Energieerhaltungssatz. Dieser besagt, dass alles Licht reflektiert wird, das auf eine Oberfläche auftrifft und nicht vollständig absorbiert wurde. Oberflächen können auch selbst Licht ausstrahlen. Dieses Verfahren ist somit unabhängig vom Betrachtungspunkt, aber sehr rechenintensiv durch hohen Zeit- und Speicherbedarf. Es ist umfangreicher in [33] beschrieben. Damit synthetische Objekte real dargestellt werden können, sollte eine globale Beleuchtung der Objekte verwendet werden. Dabei muss abgewogen werden, ob der rechenintensivere Aufwand mit einer verbesserten realen Darstellung im Verhältnis steht.

2.7.3 Beleuchtung im Cinematic Renderer

Da die synthetischen Objekte nicht vom CR gerendert werden, muss eine zusätzliche Beleuchtung für die synthetischen Objekte erstellt werden. Um die Beleuchtung der Objekte möglichst gleich der Beleuchtung vom Cinematic Renderer darzustellen, muss die Beleuchtung vom Cinematic Renderer verstanden werden. Dieser ist, wie schon im Abschnitt 2.4.1 beschrieben, eine Erweiterung zum *Path Tracing*, also ein globales Beleuchtungsmodell. Ein Unterschied zum normalen *Path Tracing* ist, dass es nicht eine Lichtquelle gibt, die beim Verlassen des Volumens zum Pixelfarbwert multipliziert wird, sondern eine Beleuchtungsmap um das Volumen liegt. Auf dieser wird beim Verlassen der Bounding Box des Strahls der Lichtfarbwert nachgeschlagen und zum momentanen

(a) Beleuchtungsmap

(b) irradiance Beleuchtungsmap

Abbildung 2.6: Vergleich Beleuchtungsmap und irradiance Beleuchtungsmap [58]

Pfadfarbwert multipliziert. Die Beleuchtungsmap kann entweder kugel- oder würfelförmig vorliegen und kann unterschiedlichste Umgebungen von Wald bis Apartment darstellen. Aufgenommen wurden die Bilder mit einer Stereokamera im *High Dynamic Range* (HDR) Format. HDR Bilder sind, wie der Name schon verrät, Bilder mit großem Farb- und Helligkeitumfang. Dadurch scheinen Bilder in diesem Format realistischer für den Anwender [32]. Eine Beleuchtungsmap oder auch *Reflections-Map* ist eine Speicherung der Umgebung in einer Textur. Diese kann dann verwendet werden, um entweder eine *Reflection Cubemap* oder eine *Diffuse Cubemap* zu erstellen. Die Beleuchtungsmap kann so wie sie ist auf die sechs Seiten eines Würfels abgebildet werden, dadurch entsteht die *Diffuse Cubemap*. Die *Reflection Cubemap* wird erzeugt, indem Vektoren aus Blickrichtung auf die Oberfläche der *Diffuse Cubemap* auftreffen und in eine neue Richtung reflektiert werden. Bei Änderung der Blickrichtung wird diese wieder neu berechnet im Gegensatz zur *Diffusen Cubemap*. Da die *Diffuse Cubemap* für eine Beleuchtung zu viele Details enthält, wird diese unscharf gemacht (engl. blurring) und die entstandene Beleuchtungsmap wird *Irradiance Cubemap* genannt. Ein Vergleich einer Beleuchtungsmap und der dazugehörigen irradiance Beleuchtungsmap ist in Abbildung 2.6 dargestellt. Die *Reflection Cubemap* sieht ähnlich aus, wie die Beleuchtungsmap mit verstärkter Intensität an den hellen Pixeln. Bei der Verwendung des CRs kann der Nutzer zwischen unterschiedlichen Beleuchtungsmaps auswählen, welche in ganz verschiedenen Umgebungen aufgenommen wurden. Die 3D Volumen werden durch verschiedene Beleuchtungsmaps unterschiedlich dargestellt. Dies hat Einfluss auf die Wahrnehmung von Tiefeninformationen von z.B. Blutgefäßen. Es könnte die Annahme getroffen werden, dass für medizinische Visualisierungen Beleuchtungsmaps, die in medizinischen Umgebungen aufgenommen wurden, am besten geeignet sind. Allerdings erweisen sich auch Beleuchtungsmaps von natürlichen Umgebungen, wie ein Wald, als geeignet, um die 3D Volumen zu beleuchten. Leider gibt es noch keine Studien über die Einflüsse, der in den unterschiedlichen Umgebung aufgenommenen Beleuchtungsmaps, auf die Wahrnehmung des 3D Volumens.

Kapitel 3

Konzept

In diesem Kapitel werden Konzepte vorgestellt, wie die Erweiterungen, die in Kapitel 1.2 beschrieben wurden, umgesetzt werden können. Es wird zuerst auf das Labeling von Objekten eingegangen, dann auf eine Methode, wie die Objekte im Volumen für den Nutzer immer sichtbar sein können. Zusätzlich wird eine Methode für die Beleuchtung der synthetischen Objekte vorgestellt.

3.1 Labeling

Die automatische Platzierung von externen Labels ist aufgrund von einigen Kriterien, die bei der Anordnung von Labels berücksichtigt werden müssen, eine komplexe Aufgabe. Die wichtigsten Anforderungen für die Platzierung der Labels sind:

- Labels dürfen sich nicht gegenseitig überlagern
- Labels müssen am äußeren Rand der Anzeige liegen
- Zwei oder mehrere Verbindungslinien von Label und Ankerpunkt dürfen sich nicht kreuzen
- Label darf nicht die am Rand präsentierten statischen Informationen überlagern

Die Anforderungen sind für 2D und 3D Bilder gleich, jedoch ist die Einhaltung der Kriterien in 3D viel komplexer. Mit der automatischen Platzierung von Labels in 2D hat sich im Jahr 2012 eine Bachelorarbeit [36] bei Siemens Healthcare GmbH befasst. Diese Implementierung kann für diese Arbeit als Basis herangezogen werden. Für die automatische Platzierung von Labels in 3D wird eine eigene Methode entwickelt. Im folgenden wird zuerst das schon bestehende Verfahren für die automatische Platzierung von Labels in 2D und dann ein eigener Ansatz für 3D dargestellt.

In der Arbeit [36] wurde ein neues Verfahren zur automatischen Platzierung entwickelt, welches echtzeitfähig ist und zufriedenstellende Ergebnisse liefert. Dabei werden die Labels nach und nach hinzugefügt, indem sie zuerst an der, nach den in der Arbeit aufgestellten Kriterien, besten Position platziert werden. Dann wird geprüft, ob sich Labels überschneiden, falls ja, werden die Label, die das neu hinzugefügte Label überdecken, nach und nach verschoben. Der genaue Ablauf dieses Verfahrens ist in der Bachelorarbeit

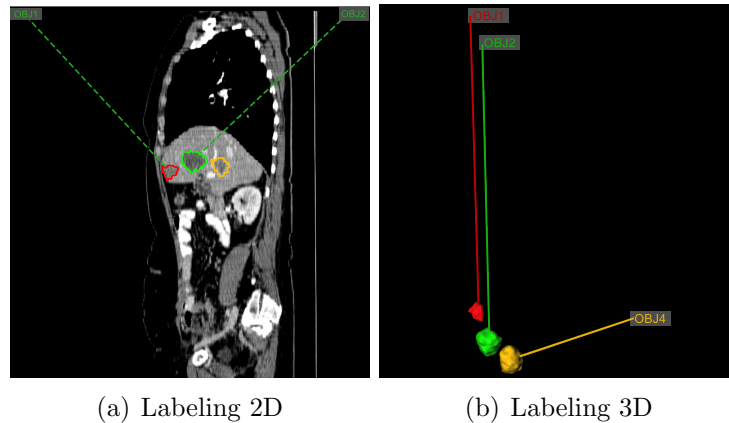


Abbildung 3.1: Vergleich Labeling 2D und 3D

beschrieben. In Abbildung 3.1(a) ist die automatische Platzierung der Labels in 2D zu sehen.

Für die Platzierung der Labels in 3D ist ein Ansatz entwickelt worden, welcher nicht immer alle aufgestellten Anforderung erfüllt, aber akzeptiert wurde. Da die automatische Platzierung von Labels ein sehr komplexes Thema ist, wurde dieses auf Grund der zeitlichen Beschränkung der Bearbeitungszeit der Bachelorarbeit nicht so hoch priorisiert wie andere Konzepte. Auf die Verbesserung des Ansatzes wird in Kapitel 6.2.2 eingegangen. Die Position der Labels wird bei Hinzufügen von einem neuem Objekt anhand er Bounding Box¹ des Volumens² berechnet und ändert sich beim Hinzufügen von weiteren Labels für die schon gelabelten Objekte nicht. Bei Änderung eines Objekts, z.B. durch eine Veränderung der Größe oder Position wird die Position des Labels von diesem Objekt neu berechnet. Für die Berechnung der Position eines Labels wird die Bounding Box in vier Teile unterteilt und geprüft, in welchem Teil das zu labelnde Objekt liegt. Je nachdem in welchem Teil der Bounding Box das Objekt liegt, wird es am äußeren Rand dieser Bounding Box in 2D projiziert. Ein Ergebnis dieser Platzierung der Labels ist in Abbildung 3.1(b) zu sehen.

3.2 Smart Visibility kombiniert mit Cinematic Rendering

In Abbildung 3.2(b) wird ein synthetisches Objekt, welches eine segmentierte Leber (grün) ist, mit dem vom CR gerenderten Volumen dargestellt. Die Segmentierung ist für den Betrachter wichtig und sollte deshalb aus jeder Perspektive sichtbar sein, allerdings kann die Sichtbarkeit nur durch eine Fensterung der Transferfunktion oder die Wahl einer anderen Transferfunktion erreicht werden, die dann wiederum das Volumen ändert und Informationen von diesem nicht abgebildet werden. Dieses kann in Abbildung 3.2 gese-

¹Eine Bounding Box ist ein geschlossenes Volumen, welches ein anderes komplett beinhaltet. Diese kann sich auf verschiedene weisen nach dem Objekt ausrichten. In diesem Fall wird eine axenbasierte Bounding Box verwendet, die sich nach dem Weltkoordinatensystem ausrichtet und durch den minimum und maximum Knoten aufgespannt wird[44].

²Mit Volumen ist das vom CR dargestellte Volumen gemeint.

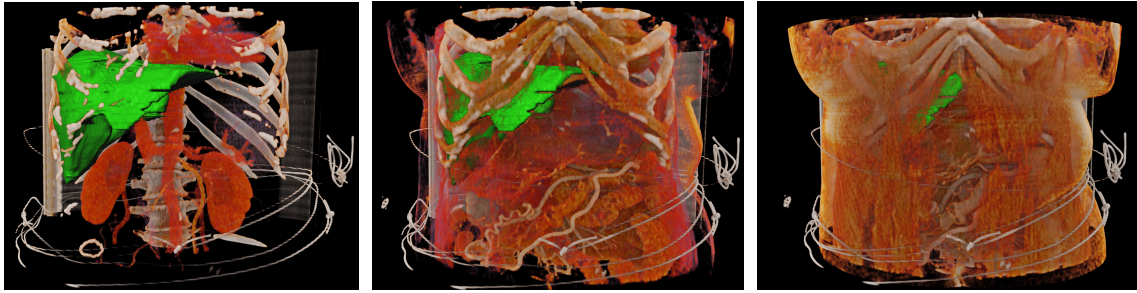


Abbildung 3.2: Segmentierung mit dem vom CR gerendertem Volumen mit verschiedenen Fensterungen

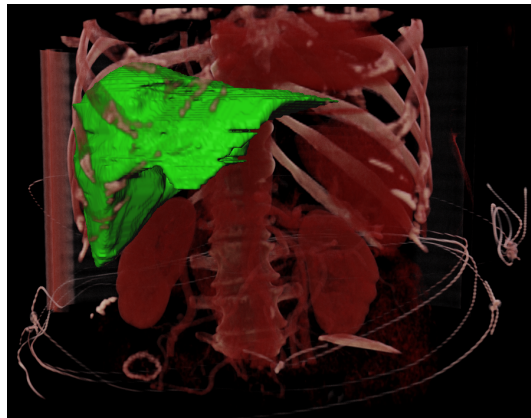
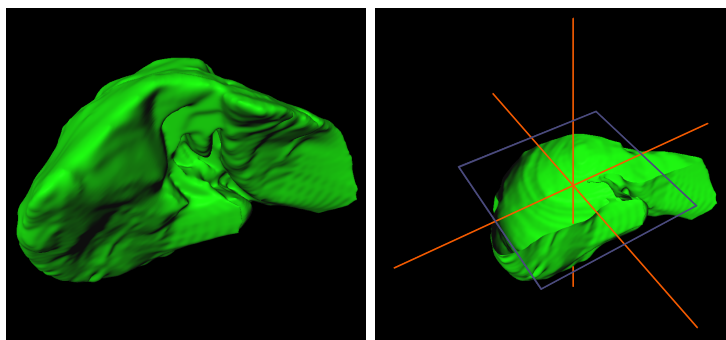


Abbildung 3.3: Segmentierung mit vom CR gerendertem Volumen mit einer anderen Transferfunktion

hen werden. Dabei ist in Abbildung 3.2 die gleiche Transferfunktion verwendet worden mit unterschiedlichen Fensterungen und in Abbildung 3.3 eine andere Transferfunktion verwendet worden, welche vorwiegend Gefäße und Organe darstellt. Damit die synthetischen Objekte immer sichtbar sind, wurde eine Methode entwickelt, die ähnlich dem im Abschnitt 2.6 genannten *Illustrative Context-Preserving Volume Rendering* ist. Zusätzlich wurde eine Methode entwickelt, um eine Schnittebene automatisch zu platzieren. Diese wird im Folgenden kurz dargestellt, gefolgt von einer ausführlichen Erläuterung über die Methode zum Ausschneiden mit einer *Clipping Textur*.

3.2.1 Schnittebene über Hauptkomponentenanalyse platzieren

Eine Schnittebene ermöglicht dem Nutzer bestimmte Bereiche vom Volumen auszublen- den, sodass die Teile des Volumens ausgeblendet werden, die auf der einen Seite der Schnittebene liegen und die Teile, die auf der anderen Seite liegen, dargestellt werden. Eine gute automatische Platzierung der Schnittebene kann dem Nutzer Zeit ersparen. Dies liegt daran, dass die Schnittebene vier Freiheitsgrade den Roll-, Nick- und Gier-Winkel (engl. yaw, pitch and roll) sowie Transformation in zwei Richtungen entlang einer Achse hat. Die Frage ist, was eine „gute“ Platzierung ist. Dafür wird sich am Verfahren der Hauptkomponentenanalyse (engl. Principle Component Analysis (PCA)) bedient. Bei diesem Verfahren werden orthogonale Basisvektoren gebildet, die in Richtung der größten Varianz der zugrunde liegenden Daten zeigen. Diese Vektoren werden nach absteigender



(a) Mesh einer segmentierten Leber ohne Schnittebene (b) Mesh einer segmentierten Leber mit Schnittebene

Abbildung 3.4: Vergleich mit und ohne Schnittebene

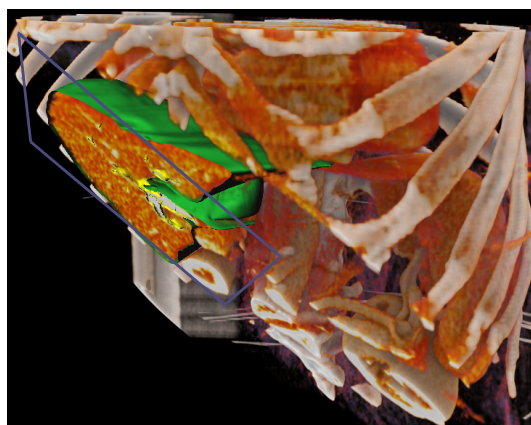


Abbildung 3.5: Segmentierung mit vom CR gerendertem Volumen und Schnittebene

Varianz angeordnet [44]. Für statistische Anwendungen wird die PCA verwendet, um Dimensionen des Merkmalsraums zu reduzieren. Hier wird sie benutzt, um die drei Vektoren zu finden, welche die drei größten Ausdehnungen der Knoten des Meshes entlang je einer Achse beschreiben. Die Punktmengen, die hier betrachtet werden, sind Masken³ der segmentierten tomographischen Daten. Die Schnittebene wird dann durch die zwei ersten Basisvektoren ausgerichtet. Die drei Basisvektoren liegen absteigend nach Varianz sortiert vor. In Abbildung 3.4(a) ist die segmentierte Leber (grün) ohne Schnittebene dargestellt und in Abbildung 3.4(b) mit automatisch platzierter Schnittebene durch PCA. Die Schnittebene ist durch das lila farbige Rechteck visualisiert. Die orangen Linien stellen die Richtungen der drei Basisvektoren dar. Eine Darstellung des vom CR gerendertem Volumen, der segmentierten Leber (grün) und den Blutgefäßen (gelb) mit Schnittebene (lila) ist in Abbildung 3.5 dargestellt.

³Segmentierungen werden häufig als Masken zwischengespeichert bevor die 3D Oberflächen aus diesen berechnet werden. Eine Maske beinhaltet alle Voxel des Ausgangsvolumens gelabelt mit einer eins oder null, je nach dem ob der Voxel zum Objekt oder zum Hintergrund gehört. Im weiteren Verlauf der Arbeit ist mit einer Maske immer die genannte Definition einer Maske gemeint.

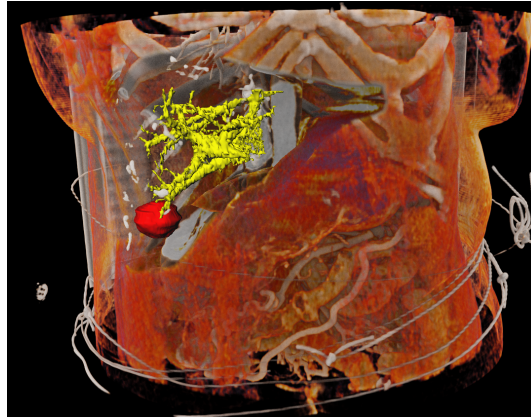


Abbildung 3.6: Segmentierung mit vom CR gerendertem Volumen

3.2.2 Ausschneiden mit *Clipping Texture*

Die entwickelte Methode dient dazu, dass die synthetischen Objekte und die Segmentierungen aus jedem Betrachtungswinkel sichtbar in der Szene sind, auch dann, wenn sie eigentlich im Volumen liegen würden oder von Teilen des Volumens verdeckt werden würden. Das genannte Problem soll durch ein Ausschneiden des Volumens mit einer *Clipping Texture* gelöst werden. Ein Resultat der im folgenden beschriebenen Methode ist in Abbildung 3.6 zu sehen. Bei diesem wurde der Bereich der segmentierten Leber aus Abbildung 3.2 ausgeschnitten, wodurch die segmentierten Blutgefäße (gelb) und ein Tumor (rot) in der Leber sichtbar wird. Dafür wird eine Textur der Szene erstellt, die dann an den CR weitergegeben und in diesem verarbeitet wird. In den folgenden Abschnitten wird zuerst die Erstellung dieser *Clipping Texture* erläutert, dann wird eine Methode beschrieben, wie die synthetischen Objekte durch andere einfache Formen, wie Kugeln oder Würfel, ersetzt werden können und Grundlage für die *Clipping Texture* bilden können. Der letzte Abschnitt stellt dar, wie die *Clipping Texture* im CR verarbeitet wird.

Erstellung der *Clipping Texture*

Der Ablauf der Erstellung der *Clipping Texture* ist in Abbildung 3.7 dargestellt und wird im Folgenden Schritt für Schritt erläutert.

Gestartet wird mit der aktuellen 3D Szene (1.), die Segmentierungen oder synthetische Objekte als Meshes enthält. Diese Meshes werden dann zu Masken konvertiert (2.), da der nächste Schritt für Masken einen geringeren Rechenaufwand benötigt als für Meshes. Was Masken sind, wurde bereits in 3.2.1 erläutert. Das Konvertieren von einem Mesh zu einer Maske ist in *MeVisLab* bereits implementiert. Die Masken werden dupliziert und an den nächsten Schritt (2.a), die Vergrößerung der Maske durch eine Dilatation⁴ weitergegeben und einmal in Originalgröße weitergeben. Durch Änderung der Größe des Strukturelements wird die Maske so verändert, dass mehr Voxel zur Segmentierung gehören, also die Segmentierung vergrößert wird. Nach Anpassung der Größe der Maske wird diese sowie die unveränderte Maske zurück als Mesh umgewandelt (3.). Dies geschieht unter Verwendung eines von *MeVisLab* implementierten *Neighboring Cells* Al-

⁴Dilatation ist eine morphologische Operation, bei der ein Strukturelement verwendet, um das Bild zu vergrößern, Löcher zu schließen und Segmente zu vereinen. [54]

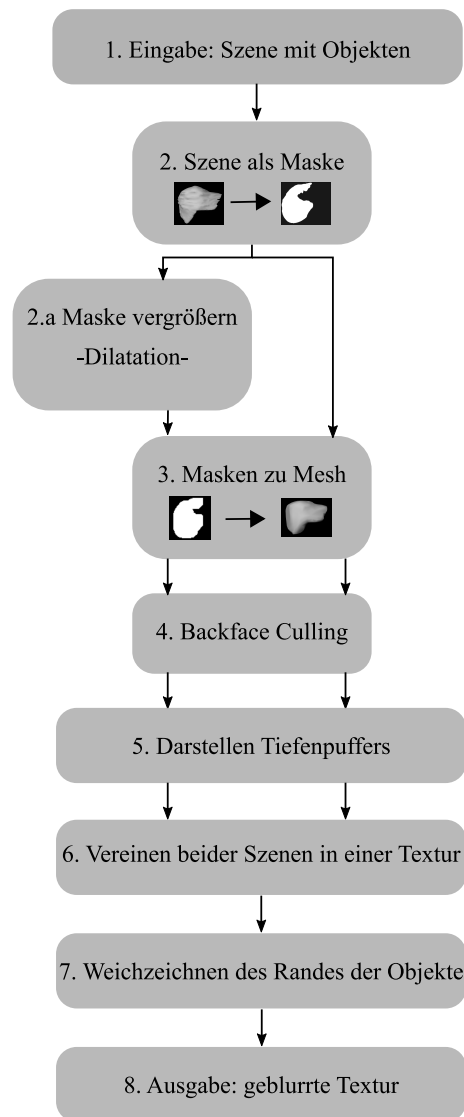


Abbildung 3.7: Ablauf Erstellung der *Clipping Textur*

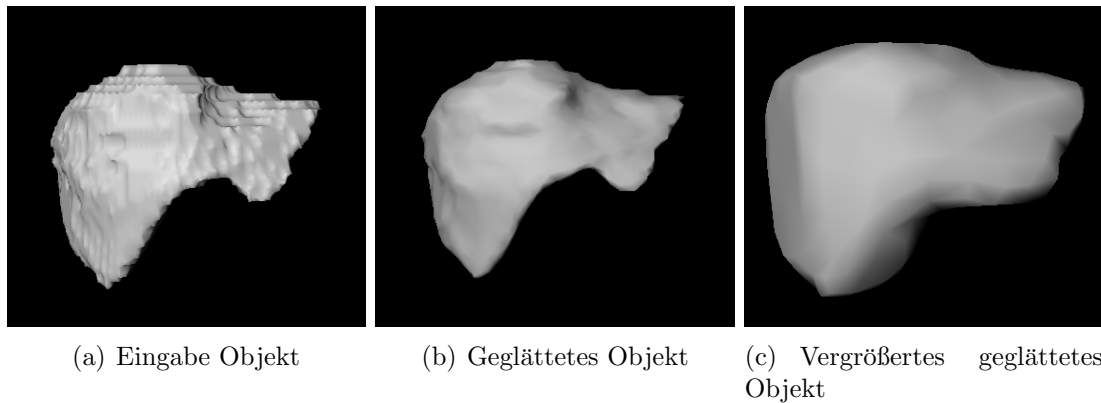


Abbildung 3.8: Vergleich der Veränderung eines Objekts für die *Clipping Textur*

gorithmus, welcher ähnlich dem *Marching Cubes Algorithmus*[31] ist. Die Oberfläche des Meshes wird dann mittels eines Laplace Operators geglättet. Zusätzlich zur Glättung werden die Polygone des Meshes um eine zu wählenden Prozentzahl reduziert, dieser Algorithmus sowie der Algorithmus zur Glättung ist ebenfalls durch *MeVisLab* gegeben. Die Reduzierung des Meshes um Polygone ist je nach Anwendung zu wählen, da es die Form des Meshes vereinfacht und dadurch Informationen des Meshes, die je nach Anwendung wichtige oder überflüssig sind, verloren gehen. In Abbildung 3.8 ist eine Segmentierung einer Leber in den unterschiedlichen Phasen der Verarbeitung zu sehen. 3.8(a) zeigt die originale Segmentierung, 3.8(b) zeigt die geglättete Segmentierung und 3.8(c) die dilatierte geglättete Segmentierung. Es ist deutlich zu erkennen, dass die Segmentierung an Detailgenauigkeit bei der Vergrößerung und Glättung verliert. Dies ist für den hier verwendeten Zweck gewollt, da die Segmentierung aus dem gerenderten Volumen ausgeschnitten werden sollen und dabei kleine Verformungen nicht berücksichtigt werden sollen.

Nachdem die Segmentierungen wieder als Meshes vorliegen, werden diese durch das *Backface Culling* (4.) weiter verändert. Das *Backface Culling* ist ein Verfahren, um die Performance bei der Darstellung von 3D Objekten zu verbessern, indem nur die Flächen verarbeitet werden, die vom Betrachter gesehen werden. Dazu wird für jedes Polygon getestet, ob die Normale in Blickrichtung zeigt. Falls ja, wird das Polygon weiter in der Grafikpipeline verwendet. Vorher muss noch festgelegt werden, wie die Knoten der Polygone nummeriert sind. Diese können entweder im Uhrzeigersinn oder gegen den Uhrzeigersinn nummeriert werden, was die Stirnfläche der Polygone und somit auch die Richtung der Normalen bestimmt [44]. Für die Erzeugung der *Clipping Textur* wird das *Backface Culling* genau andersherum verwendet. Es werden nur noch die Polygone angezeigt, die für den Betrachter nicht sichtbar sind, also deren Normale vom Betrachter weg zeigt. Ein Beispiel, wie das *Backface Culling* eines Würfels aussieht, ist in Abbildung 3.9 zu sehen. Durch das *Backface Culling* werden nicht bei allen Meshes alle Polygone ausgeblendet, die für die *Clipping Textur* ausgeblendet werden sollen. Bei Meshes, die nicht vollständig konvex sind, so wie die segmentierte Leber aus Abbildung 3.8, bleiben Polygone sichtbar, die zwischen den hintersten Polygonen des Meshes und dem Betrachter liegen. Dies kann nicht durch das *Backface Culling* gelöst werden, sondern es müsste ein anderes Verfahren verwendet werden. Um wirklich nur die hintersten Polygone eines Meshes sichtbar zu machen, müsste die Reihenfolge des Renderings invertiert werden.

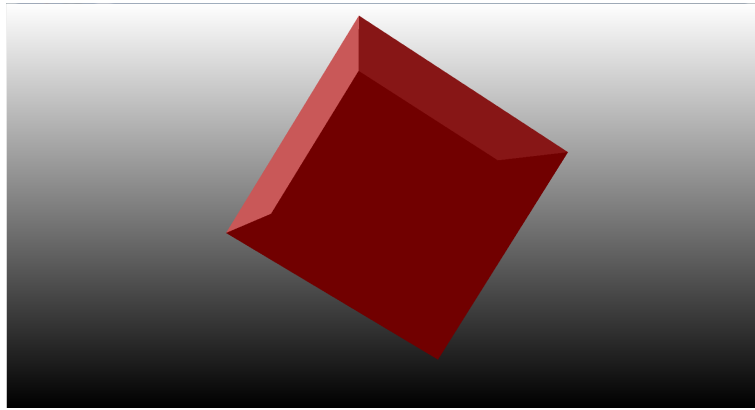


Abbildung 3.9: Backface Culling Würfel

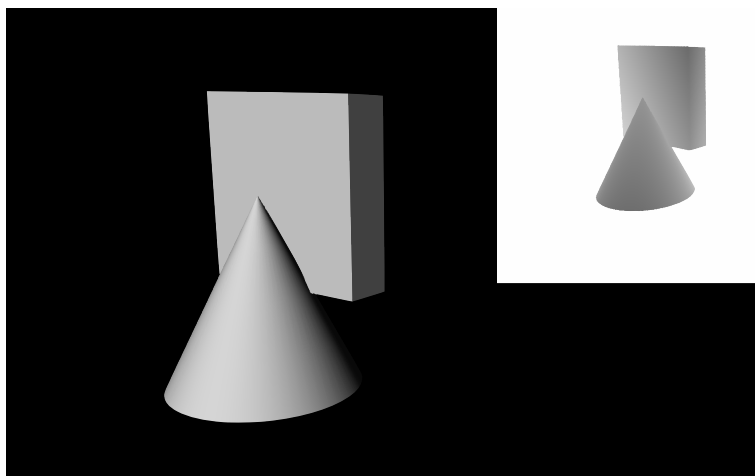


Abbildung 3.10: Tiefenpuffer einer Szene

Ein Algorithmus der genau dies macht, ist der *Reverse Painter's* Algorithmus. Bei diesem werden Objekte, die den geringsten Abstand zum Betrachter haben, als erstes gezeichnet [46]. Somit würden die Objekte die ganz hinten liegen, als letztes auf das resultierende 2D Bild gezeichnet werden. Dies ist allerdings in der verwendeten Software (MeVisLab 2.8.2) derzeit nicht möglich.

Der nächste Schritt (5.) ist das Darstellen der Tiefenwerte der Meshes. Dafür wird sich am Tiefenpuffer der Szene bedient. Der Tiefenpuffer oder auch *z-Puffer* ist eine Textur, in der die Tiefenwerte als Distanzen zu Objekten der 3D Szene speichert. Ein Beispiel für den Tiefenpuffer ist in Abbildung 3.10 in der oberen rechten Ecke dargestellt. Um den Tiefenpuffer zu erzeugen, werden zuerst alle Einträge des Puffers (alle Pixel) mit unendlich gleichgesetzt, dies steht für den größtmöglichen Abstand zum Betrachter. Dann werden die Polygone der Szene abgetastet und in die Pixel des Bildes übertragen, aber nur dann, wenn der Wert des Pixels kleiner als der *z*-Wert des aktuell betrachteten Polygons ist [22]. In Abbildung 3.11 sind in der oberen rechten Ecke die Tiefenwerte der Segmentierung der Leber zu sehen. Je näher ein Polygon zum Betrachter ist, desto kleiner ist der Abstand und demnach auch der Tiefenwert. Dadurch werden die Polygone, die eine kleinere Distanz zum Betrachter haben, dunkler dargestellt und die, die weiter weg sind, heller.

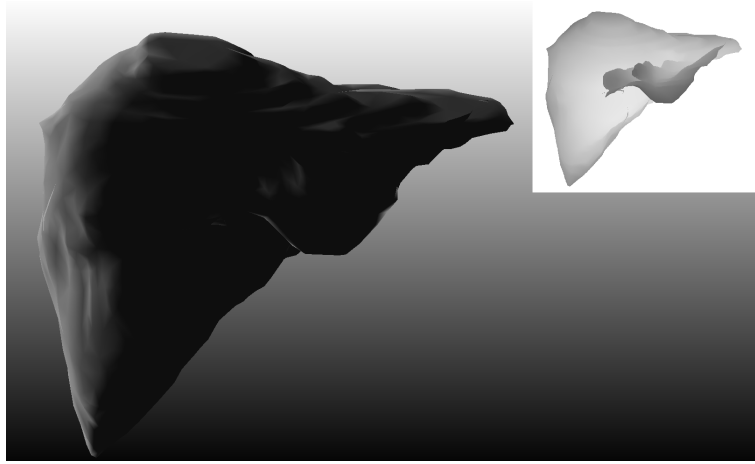


Abbildung 3.11: Tiefenpuffer eines Objekts

Im nächsten Schritt (6.) werden die beiden vorliegenden Szenen zu einer Textur vereint. Dafür werden die zwei erzeugten Szenen verarbeitet. Die Textur, die entsteht, enthält Informationen über die Szene. Dabei werden die Informationen über die gegebenen Szenen in den drei Farbkänen und dem α -Kanal gespeichert. Dadurch sind die Informationen vom CR getrennt ausgelesen und können weiterverarbeitet werden. In Abbildung 3.12 sind die Bilder, die pro Farbkanal abgespeichert sind, dargestellt. Im roten Kanal werden die Informationen über die unvergrößerte Szene gespeichert. Es ist zu beachten, dass die gespeicherten Werte die Tiefeninformationen über die Szene sind, folglich ist z.B. in Abbildung 3.12(a) das Mesh dunkelrot und alles was nicht zum Mesh gehört *reines*⁵ Rot. Im grünen Farbkanal 3.12(b) werden die Informationen über die vergrößerte Szene gespeichert und im blauen Farbanal 3.12(c) wird mit einem binären Wert gespeichert, ob ein Pixel zum Mesh gehört oder nicht. Als Mesh wird dabei die vergrößerte Szene verwendet. Im α -Kanal werden Informationen über den Rand des vergrößerten Meshes gespeichert. Dieser wird im nächsten Schritt erst noch weichgezeichnet, was im nächsten Unterabschnitt genauer erklärt wird. Eine Darstellung aller Farbkäle der Textur ist in Abbildung 3.13 zu sehen. Die Farben dieser Darstellung müssten eigentlich genau invers betrachtet werden. Der gelbe Bereich enthält keinen Blauanteil, da dieser nur Eins ist, wenn ein Pixel zum Mesh gehört. Dadurch besteht dieser nur aus rot und grün, was die Farbe gelb ergibt. Der blaue Bereich enthält vorwiegend blau, da im blauen Kanal dort Einsen stehen und im roten und grünen Kanal kleine Werte an diesen Stellen sind. Der hell rosa Bereich hat im blauen Kanal nur Einsen und im roten Kanal auch, weshalb kaum grüne Farbe an diesen Stellen zu sehen ist. Diese Textur wird nun in einem weiteren Schritt (7.) an den Kanten des Meshes weicher gezeichnet.

Weichzeichnen des Randes der Textur

Damit beim Darstellen der Textur im CR ein weicher Übergang von der *Clipping Textur* zum restlichen Volumen entsteht, wird der Rand der *Clipping Textur* verwischt. Die harte Kante kann verschwinden, wenn die Farbe zum Rand des Objekts verblasst, und so der Rand der Szene weicher gezeichnet wird. Das Weichzeichnen des Randes kann

⁵Mit *reinem* Rot ist Rot mit den RGB-Werten (1,0,0) gemeint.

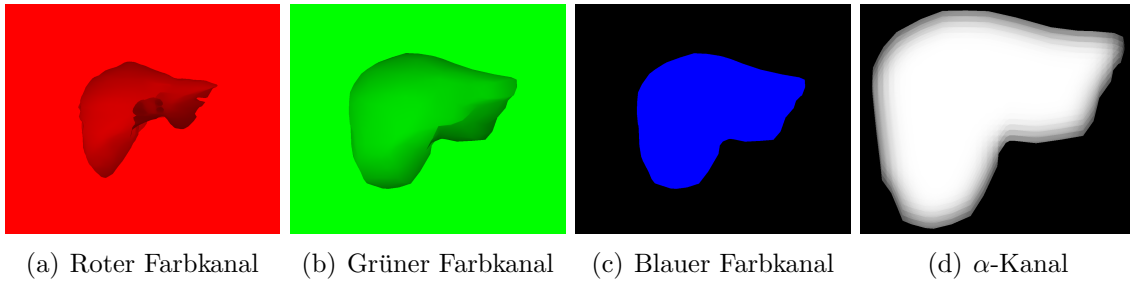


Abbildung 3.12: Vergleich Kanäle der *Clipping Textur*



Abbildung 3.13: Komplette *Clipping Textur*

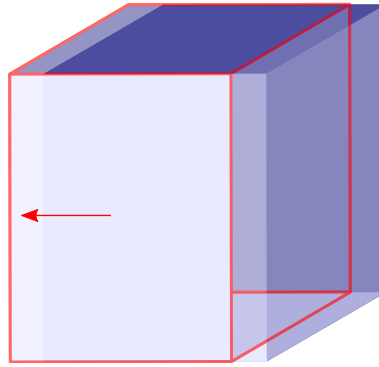


Abbildung 3.14: Beispiel Glättung von Kanten 3D

erreicht werden, indem das Bild am Rand mit abnehmender Opazität nach außen verschoben wird. Dies ist in Abbildung 3.14 gezeigt. Der Würfel ohne Umrandung wird nach links verschoben⁶ und die Opazität des Würfels auf 60% gestellt, dadurch entsteht ein weicherer Übergang von der linken Kante des Würfels und der Szene. Damit dieser an allen Kanten des Würfels entsteht, wird der Würfel in alle Richtungen nach außen verschoben. Die Häufigkeit der Verschiebung in die selbe Richtung und die Reduzierung der Opazität haben Einfluss auf die Glättung der Kanten. Damit der Übergang weich wird, wird das Objekt mit abnehmender Opazität wiederholte Male in die gleiche Richtung verschoben. Die Abnahme der Opazität entspricht den Gewichten eines Gauß-Filters. Die Länge der Verschiebung kann vom Nutzer manuell eingestellt werden. Je größer der Wert der Länge, desto größer wird der verschwommene Rand des Objekts. Ein Ergebnis dieser Glättung ist in Abbildung 3.12(d) dargestellt. Die Grauwerte, die bei der Glättung entstehen, werden im α -Kanal des resultierenden Bildes gespeichert und können dadurch getrennt im CR verarbeitet werden. Das Weichzeichnen des Randes ist in Abbildung 3.13 nicht zu sehen, da Transparenz nicht ohne Hintergrundfarbe darzustellen ist.

Ersetzen der synthetische Objekte durch einfachere Primitive

Die synthetischen Objekte sind die Grundlage der *Clipping Textur*. Der Ort, an dem ein Objekt liegt, ist von Bedeutung und soll deshalb mittels der *Clipping Textur* ausgeschnitten bzw. anders dargestellt werden. Da die Objekte teilweise sehr klein sind oder ihre Oberflächen sehr detailreich sind, ist es in manchen Fällen sinnvoll die Form der Objekte zu ändern und sie zu vergrößern. Für das Ausschneiden der Objekte aus dem Volumen, sind detailreiche Formen nicht wichtig und diese würden durch das Weichzeichnen des Randes zum Teil verschwinden. Deshalb sind Formen mit konvexen Oberflächen und weniger Details von Vorteil. Deshalb können die Objekte durch Primitive ersetzt werden, wie es auch Kubisch et al. in [26] macht. Zur Auswahl stehen vier Primitive (Kugel, Würfel, Zylinder und Kegel), die durch die verwendete Software gegeben sind. Das Primitiv, das das Objekt ersetzen soll, kann vom Benutzer ausgesucht werden. Der Ablauf der Erstellung der *Clipping Textur* ändert sich, wenn das Mesh durch eine anderes Primitiv ersetzt wird. In Abbildung 3.15 ist der Abschnitt zu sehen, der sich bei der Verwendung eines Primitiv anstelle des Objekts für eine *Clipping Textur* ändert. Im

⁶In dem Beispiel wird der Würfel nur einmal verschoben, in der Implementierung wird das Objekt 11 Mal in dieselbe Richtung verschoben.

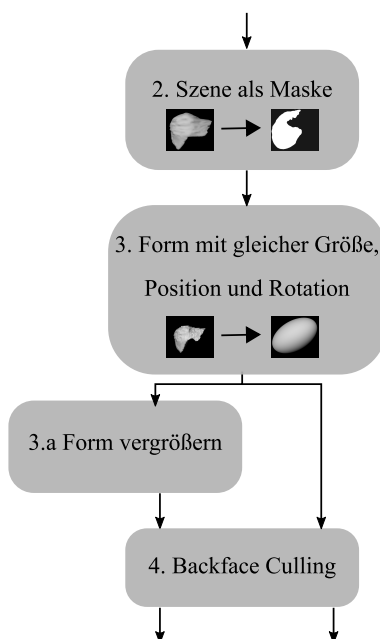
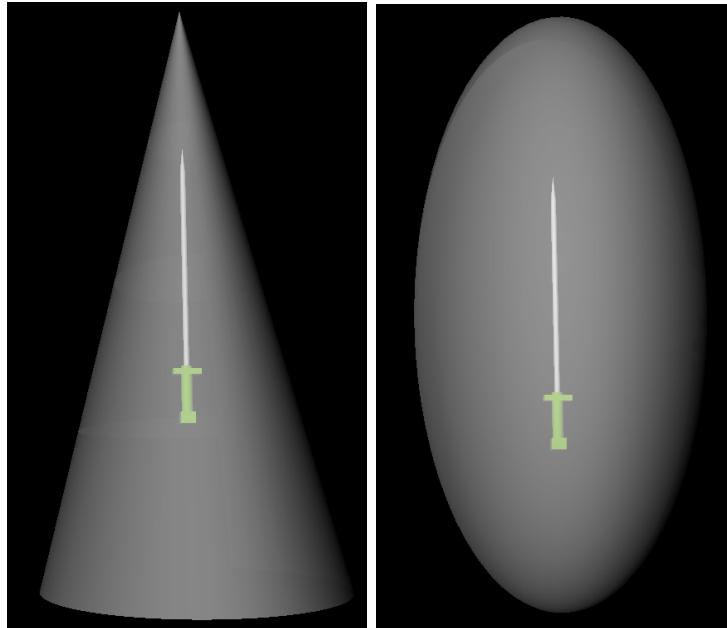


Abbildung 3.15: Ablauf Erstellung der *Clipping Textur* mit anderer Form

Ablauf ändern sich Schritt 3 und 3a. Im 3. Schritt wird die Größe des Objekts, die Position und die Rotation des zu ersetzenden Objekts ermittelt. Dafür wird die PCA, welche in Kapitel 3.2.1 schon beschrieben wurde, über die Knoten des Objekts durchgeführt. Die Rotation des Objekts in der Szene kann über die Basisvektoren der PCA berechnet werden. Für die Größe des neuen Primitives werden die Eigenwerte [2] verwendet. Die Position des neuen Primitives ist das Baryzentrum [1] der PCA. Objekte mit deutlich unterschiedlich langen Seiten, z.B. eine OP Nadel, sollen durch Formen mit ausgeglicheneren Seitenlängen ersetzt werden. Dazu wird die Skalierung der Formen verändert, indem die längste Seite bestimmt und die anderen Seiten im Verhältnis zu dieser vergrößert. Die angepasste Form wird dann noch dupliziert, damit sie dann in Schritt 3.a einmal vergrößert werden kann. Diese Vergrößerung ist abhängig vom Volumen, welches vom CR gerendert wird. Je größer das Volumen ist, desto weiter wird die Form vergrößert. Die Originalform und die vergrößerte Form wird dann weiter gegeben an Schritt 4., das *Backface Culling*. Das Ersetzen einer Nadel durch zwei unterschiedliche Primitive ist in Abbildung 3.16 dargestellt.

Verarbeitung der Textur im Cinematic Renderer

In diesem Abschnitt wird die Verarbeitung der vorher erstellten *Clipping Textur* im CR vorgestellt. Die Verarbeitung der *Clipping Textur* ist ein Ergänzung zum bestehenden Ablauf des CR, und wird nur durchgeführt, wenn vorher eingestellt wurde, dass das erste Volumen mit der *Clipping Textur* ausgeschnitten werden soll. Eine schematische Darstellung der Verarbeitung im CR ist in Abbildung 3.17 zu sehen. Diese zeigt eine Seitenansicht auf die Szene. Der dunkel-rote Bereich ist das Volumen, welches über eine frei gewählte Transferfunktion dargestellt wird. Der blaue Bereich ist das zweite Volumen: ein synthetisches Objekt, welches für den Betrachter immer sichtbar sein soll. Dieser wird mit einer zweiten Transferfunktion dargestellt. Das erste Volumen, welches eigentlich

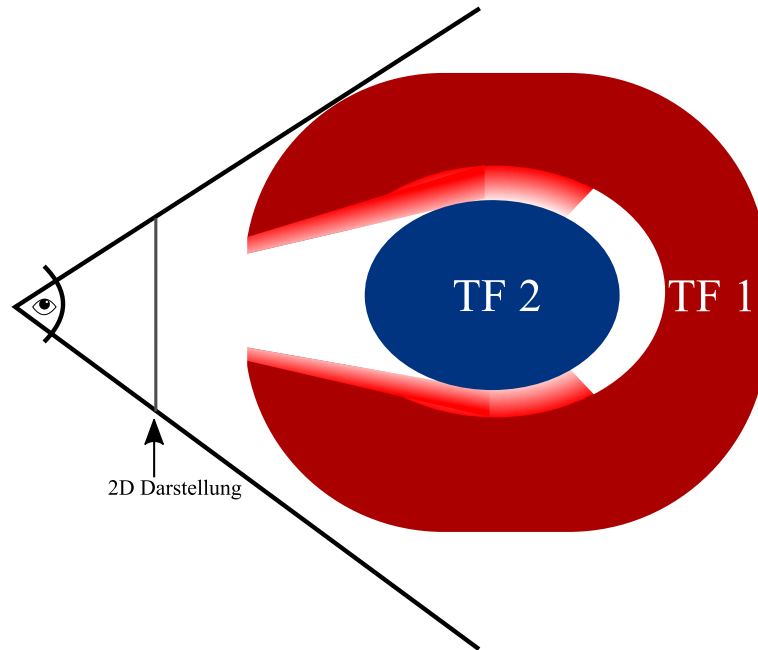
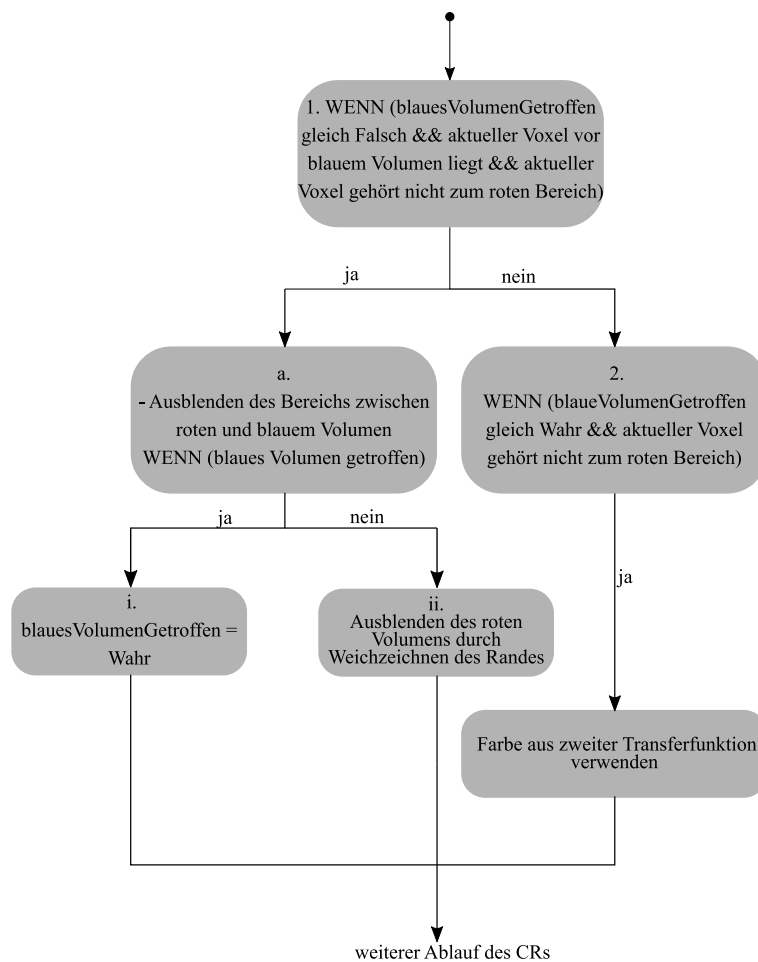


(a) Nadel ersetzt durch Kegel (b) Nadel ersetzt durch einen Ellipsoid

Abbildung 3.16: Ersetzen einer Nadel durch unterschiedlich geformte Objekte

zwischen dem Objekt und dem Betrachter liegt, wird durchsichtig dargestellt, allerdings nur an den Stellen, wo das erste Volumen das Objekt für den Betrachter verdecken würde. Im CR wird anhand der Werte der vier Kanäle der *Clipping Textur* geprüft, ob ein Pixel zum blauen Objekt oder zum roten Volumen gehört. In Abbildung 3.18 ist der Ablauf dargestellt, mit dem entschieden wird, zu welchem Volumen ein Pixel des resultierendem Bildes gehört. Der Ablauf ist in der folgende Auflistung erläutert. Für jeden Pixel des resultierendem Bildes wird geprüft, zu welchem Volumen er gehört und in welcher Transferfunktion er angezeigt werden muss.

1. Als erstes wird geprüft, ob das zweite Volumen (*Clipping Textur*) schon einmal getroffen wurde, was bedeuten würde, dass der Wert für diesen Voxel auf der *Clipping Textur* im grünen Farbkanal den Wert 1 hat. Der grüne Farbkanal gibt an, ob der Pixel zum Objekt gehört (Wert < 1) oder zur Umgebung (Wert $= 1$). Zusätzlich wird die Strahlrichtung überprüft. Dies ist in Abbildung 3.19 dargestellt. Für jeden getroffenen Voxel P wird geprüft, ob dieser vom Betrachter aus gesehen hinter dem zweite Volumen liegt oder davor. Dafür wird die initiale Strahlrichtung i , die zu diesem Voxel geführt hat, mit der Richtung des Vektors v vom aktuellen Voxel im ersten Volumen zum aktuellem Voxel im zweiten Volumen verglichen. In Abbildung 3.19(a) liegt der aktuell betrachtete Voxel hinter dem zweiten Volumen. Mathematisch kann dies durch das Berechnen des Skalarprodukts der beiden Vektoren i und v geprüft werden. Ist das Skalarprodukt größer als null, bedeutet es, dass der Winkel zwischen den Vektoren kleiner als 90° ist und somit der Voxel vor dem zweiten Volumen ist. Wenn nun das zweite Volumen noch nicht getroffen wurde und auch der aktuelle betrachtete Voxel nicht hinter dem zweiten Volumen liegt, geht es mit Schritt (a) weiter, falls das zweite Volumen schon getroffen wurde geht es mit Schritt 2. weiter.

Abbildung 3.17: Verarbeitung der *Clipping Texture* im CRAbbildung 3.18: Flussdiagramm zur Verarbeitung der *Clipping Texture* im CR

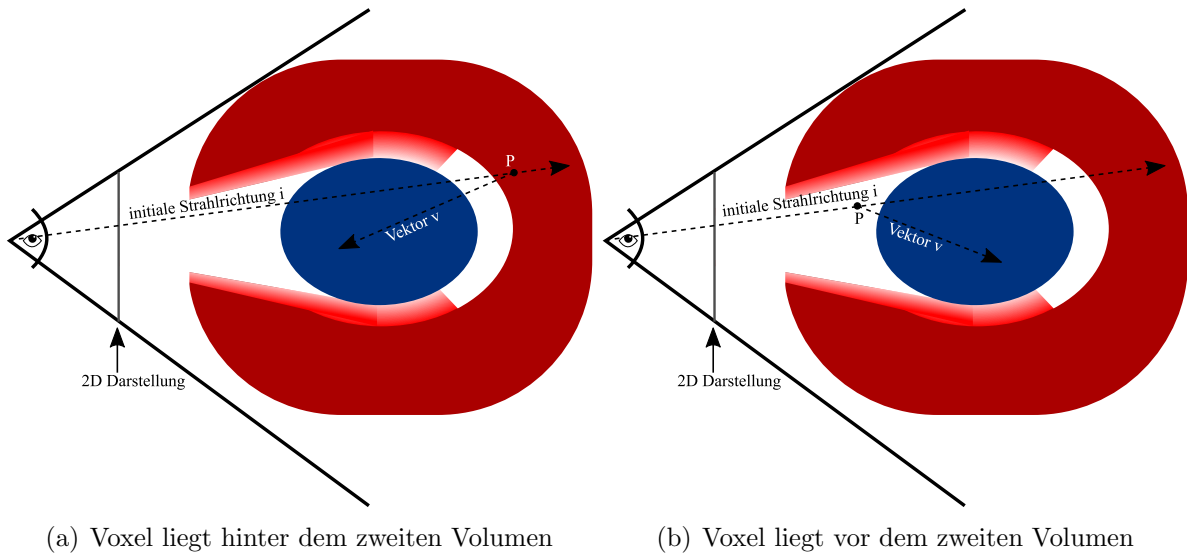


Abbildung 3.19: Test: liegt Voxel hinter dem zweiten Volumen

(a) In diesem Schritt wird der Teil des roten Volumens, der vor dem blauen liegt ausgeblendet. Allerdings nur solange, bis das zweite Volumen getroffen wird. Wenn dieses getroffen wird, geht es in Schritt i. weiter. Ansonsten wird das Ausblenden in Schritt ii. umgesetzt.

- i. Wenn das zweite Volumen getroffen wird, wird bei Verarbeiten des nächsten Pixels nicht mehr in Schritt 1. gegangen, sondern direkt geprüft, ob die Bedingung für Schritt 2. erfüllt ist.
- ii. Wenn der aktuelle Voxel noch vor dem zweiten Volumen ist, wird das erste Volumen ausgeblendet. Dies passiert unter Verarbeitung des α -Kanals der *Clipping Textur*, welcher die Werte der Tiefentextur mit weich gezeichneten Rand beinhaltet. Der α -Wert des aktuell betrachteten Pixels wird mit dem α -Wert der *Clipping Textur* multipliziert. Dies sorgt dafür, dass der Übergang vom ausgeschnittenen Volumen zum zweiten Volumen nicht so hart erscheint.

2. Dieser Schritt wird durchgeführt, wenn das zweite Volumen schon getroffen wurde und sich der aktuell betrachtete Voxel im zweiten Volumen befindet. Was bedeutet, dass der Wert für diesen Voxel auf der *Clipping Textur* im grünen Farbkanal einen Wert kleiner als eins hat. Ist dies der Fall, wird für die Farbe des aktuell betrachteten Pixels der Wert der zweiten Transferfunktion gespeichert.

Dieser Ablauf wird in den bestehenden Ablauf des CR, welcher in Abschnitt 2.4.1 erklärt wurde, eingebunden. Dadurch wird das erste Volumen an den Stellen, an denen das zweite Volumen nicht sichtbar sein soll, wie gewohnt dargestellt. Ein Ergebnis der Verarbeitung im CR der *Clipping Textur* ist in Abbildung 3.20(a) zu sehen. Die für das Clipping verwendete Textur ist in Abbildung 3.20(c) und 3.20(b), welches das ausgeschnittene Volumen mit der überlagerten *Clipping Textur* zeigt, zu sehen.

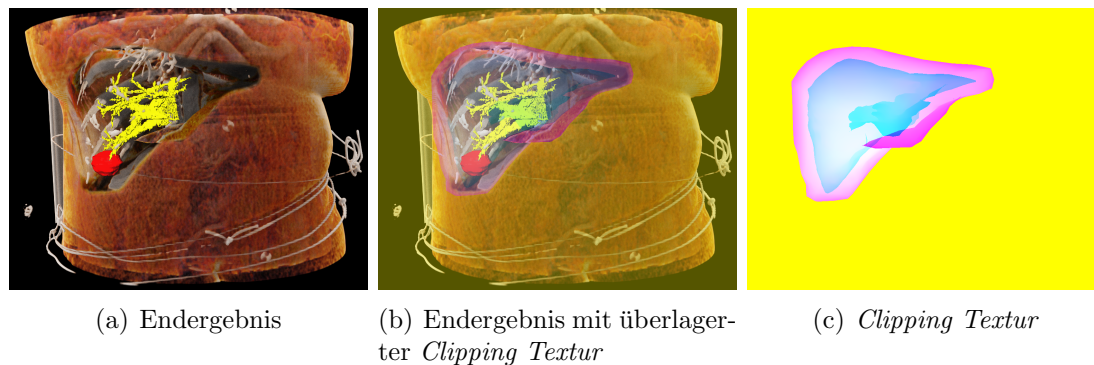


Abbildung 3.20: Clipping der segmentierten Leber

3.3 Beleuchtung der synthetischen Objekte

Die synthetischen Objekte, wie in 2.2 schon beschrieben, liegen als Meshes vor und werden nicht direkt vom CR gerendert. Damit die Objekte für den Betrachter realer in die 3D Szene eingebunden werden, wird die Beleuchtung des CR simuliert. Dafür wird im Folgenden eine Idee vorgestellt, wie die Beleuchtung des CRs simuliert werden kann und dazu zwei Lösungen präsentiert.

3.3.1 Simulation mit mehreren direkten Lichtquellen

Die Idee für die im folgenden beschriebene Beleuchtung stammt von Shahlai et al. [49], welche ein Beleuchtungsmodell entwickelt haben, um Gesichter aus Bildern zu erkennen, zu segmentieren und zu rekonstruieren. Dies ist schwierig, da die Bilder unter verschiedensten Beleuchtungen aufgenommen wurden und auch Objekte, die nicht auf dem Bild zu sehen sind, die Beleuchtung des Gesichtes beeinflussen können. Für die Rekonstruktion der Gesichter muss deshalb die Beleuchtungssituation, die bei Aufnahme des Bildes herrschte, simuliert werden. Dafür wird das Bild an m Punkten abgetastet und m direkte Lichtquellen auf einer Kugel platziert. Die Lichtquellen erhalten den Farbwert des abgetasteten Pixels und dessen Intensität. Alle Lichtquellen zusammen ergeben schließlich die Beleuchtung für das rekonstruierte Bild.

Die Umsetzung der Idee aus [49] in Kombination mit dem CR wird nun beschrieben. Der CR wird anhand einer 2D Beleuchtungsmap beleuchtet. Für die Beleuchtung der synthetischen Objekte kann die Beleuchtungsmap abgetastet werden, und an den abgetasteten Punkten direkte Lichtquellen in der Farbe der abgetasteten Punkte platziert werden. Die Lichtquellen können entweder in Form eines Würfels oder einer Kugel um die Objekte platziert werden und ist in Abbildung 3.22(a) und (c) dargestellt. Das Abtasten der Beleuchtungsmap kann auf verschiedene Weisen gelöst werden. Zwei Möglichkeiten werden im folgenden erläutert und werden in Abschnitt 5 auf ihren Einfluss auf die Beleuchtungsqualität bewertet.

Abtasten gleich großer Bereiche

Eine Möglichkeit die Beleuchtungsmap abzutasten ist das gleichmäßige Abtasten mithilfe eines Gitters. Dazu wird die Cubemap in gleich große Bereiche aufgeteilt, wie in Bild

Abbildung 3.21: Abtasten der Beleuchtungsmap mit gleichmäßigem Gitter (nach [58])

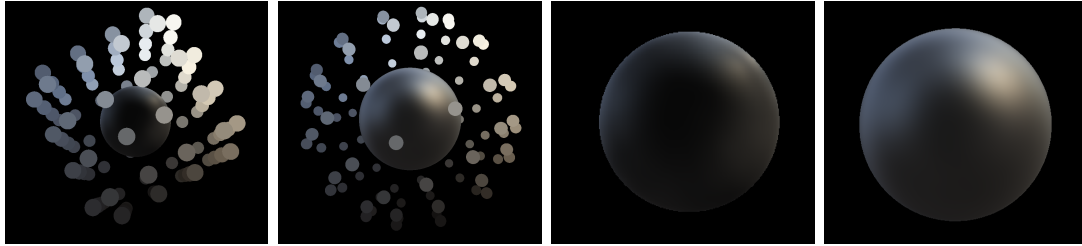
3.21 zu sehen ist. Nun wird für jedes Rechteck die Summe der Pixelfarben gebildet und durch die Anzahl der im Rechteck enthaltenen Pixel dividiert. Das Ergebnis ist der durchschnittliche Farbwert pro Rechteck. Für die Beleuchtung der Objekte werden direkte Lichter genau an den Mittelpunkten dieser Rechtecke platziert. Sie bekommen den errechneten durchschnittlichen Farbwert. Alle Lichter erhalten die gleiche Intensität.

Abtasten der drei hellsten Bereiche

Eine zweite Möglichkeit für die Beleuchtung ist die Platzierung von nur drei Lichtquellen. Für den CR gibt es einen Modus, bei dem für die Beleuchtung des Volumens drei Lichtquellen anstatt der Beleuchtungsmap verwendet werden. Diese drei Lichtquellen können die gleiche Positionen erhalten, wie die im folgenden abgetasteten Lichtquellen. Die Wahl von wenigen, für die Beleuchtung aber entscheidenden Lichtquellen, bringt Vorteile für die Rechenleistung sowie für die im Ausblick 6.2.2 beschriebene Erweiterung.

Das Abtasten der drei hellsten Pixel⁷ der Beleuchtungsmap ergibt drei Pixel, die alle nah beieinander liegen. Da dies für die Beleuchtung der Objekte bedeuten würde, dass die drei Lichtquellen nah beieinander liegen, können für die Beleuchtung mit drei Lichtquellen nicht die drei hellsten Pixel verwendet werden. Für die Wahl der drei Pixel ist die Position der Pixel und somit der Lichtquellen in 3D entscheidend. Deshalb wird jede Würfelseite der Beleuchtungsmap abgetastet und der hellste Pixel pro Seite abgespeichert. Das Abtasten ergibt sechs Pixel mit je einem Pixel pro Würfelseite. Die drei hellsten Pixel dieser sechs werden dann ausgewählt und als direkte Lichtquellen in 3D platziert.

⁷Der hellste Pixel ist der Pixel, welcher den größten Grauwert hat, also die Summe der RGB-Werte geteilt durch drei.



(a) Lichtquellen in Form eines Würfels platziert (b) Lichtquellen in Form einer Kugel platziert (c) Lichtquellen in Form eines Würfels um die Kugel (d) Lichtquellen in Form einer Kugel um die Kugel

Abbildung 3.22: Formen der Platzierung der Lichtquellen

3.3.2 Platzierung der Lichter in 3D

Um die Objekte in der Szene zu beleuchten, müssen die Lichter um die Objekte herum platziert werden. Dafür kann die Beleuchtungsmap, welche eine Cubemap ist, zu einem Würfel rekonstruiert werden. Die Gitterpunkte, auf welchen die Lichter platziert sind, werden abhängig davon, auf welcher Seite des Würfels sie liegen, in eine Richtung verschoben. Die ist in Abbildung 3.22(a) zu sehen.

Eine andere Idee die Lichter in 3D zu platzieren ist, sie als eine Kugel anzuordnen. Für jede Lichtquelle wird die Position $vecPosLight$ berechnet, indem der Ortsvektor $vecGridpoint$ zu jedem Gitterpunkt normiert wird und mit dem Radius r multipliziert wird. Dies ist in Gleichung 3.1 dargestellt.

$$vecPosLight = r \cdot normalize(vecGridpoint). \quad (3.1)$$

Kapitel 4

Implementierung

Im folgenden Kapitel wird zuerst die Hard- und Software, die für die Umsetzung der Arbeit verwendet wurde, beschrieben, und dann der Prototyp, der im Rahmen der Arbeit erweitert wurde, vorgestellt.

4.1 Hard- und Software

Nach einer kurzen Darstellung, welche Hardware und Systeme für die Entwicklung eingesetzt wird, wird die, für die Implementierung verwendete Software beschrieben.

4.1.1 Hardware und Systeminformationen

Für die Implementierung wurde ein PC mit einem Intel Xeon Prozessor E5-1620 v2 Prozessor der Taktung 3.70 GHz verwendet, der einen Arbeitsspeicher von 8 GB hat und ein 64 Bit-Betriebssystem von Windows 7 hat. Außerdem ist die Quadro K2000 Grafikkarte von NVIDIA eingebaut und es ist die Treiberversion 353.90 installiert.

4.1.2 Software

Als Programmierumgebung wird MeVisLab Version 2.8.2 und Microsoft Visual Studio Professional 2013 (VS) verwendet. MeVisLab ist eine Forschungs- und Entwicklungsumgebung um Algorithmen für die Bildverarbeitung, zur Bildanalyse und für Visualisierungen zu entwickeln. Das Hauptaugenmerk liegt dabei auf der Verarbeitung von medizinischen Daten. Es ist eine graphische Programmierumgebung mit der Prototypen und deren Benutzeroberflächen entwickelt werden können. Die Software wird von Fraunhofer MEVIS und Mevis Medical Solutions AG stetig weiterentwickelt und stellt viele Module bereit. Außerdem ist es möglich eigene Module in C++ oder Python zu entwickeln und diese in Netzwerke anzuordnen. Zusätzlich sind einige externe Bibliotheken in MeVisLab integriert, wie zum Beispiel Open Inventor, OpenGL und ITK [15]. Es gibt drei grundsätzliche Arten von Modulen, die entwickelt werden können und von denen für die vorliegende Arbeit auch Gebrauch gemacht wurde. Diese drei sind:

- ML Module (blau): werden in C++ programmiert und werden hauptsächlich für die Bildverarbeitung eingesetzt

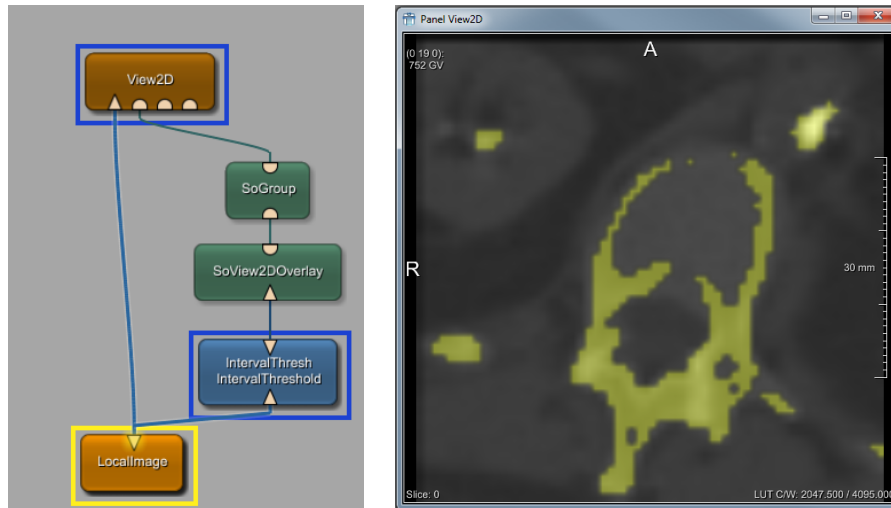


Abbildung 4.1: Beispielnetzwerk MeVisLab

- Open Inventor Module (grün): werden auch in C++ programmiert und dienen zur 3D Visualisierung und Interaktion mit der Szene
- Macro Module (braun): fassen (komplexe) Netzwerke zusammen und ermöglichen die Erstellung von graphischen Benutzeroberflächen

Für die Programmierung von C++ Modulen wird die Entwicklungsumgebung VS verwendet. Es ist möglich, die entwickelten Module mit VS zu debuggen. Die GUI wird in MeVisLab mit der MeVisLab Definition Language (MDL) programmiert. Mit dieser werden Definitionen und Skripte für alle genannten Module erstellt sowie GUIs entwickelt. MDL ist keine Programmiersprache, sondern eine Sprache, mit der Konfigurationen und Layouts festgelegt werden. MDL Dateien können in dem von MeVisLab mitgelieferten Texteditor MeVisLab Advanced Texteditor (MATE) bearbeitet werden. Die Python Skripte können auch in MATE entwickelt und getestet werden [35]. In Abbildung 4.1 ist ein einfaches Beispielnetzwerk zu sehen, welches alle Modultypen von MeVisLab enthält. Das Netzwerk lädt ein Bild und färbt die Pixel, deren Grauwerte zwischen zwei angegebenen Schwellenwerten liegen, gelb ein.

4.2 Prototyp

Der *3D Printing Prototype* ist eine von Siemens Healthcare GmbH entwickelte Software zur Erstellung von virtuellen 3D Objekten aus tomographischen Bildern. Die Software bildet das Grundgerüst der Arbeit und wird um Visualisierungs- sowie Interaktionsmöglichkeiten erweitert.

4.2.1 Grundgerüst

Die Hauptfunktion des *3D Printing Prototype* sind das Laden, Erstellen von Segmentierungen und Visualisieren von tomographischen Bildern. Es gibt zwei verschiedene Ansichtsmöglichkeiten, die eine ist für einen Monitor und ist in vier Fenster unterteilt.



Abbildung 4.2: Prototyp vor Erweiterungen

Drei der vier Fenster beinhalten dann die drei Ansichten der Multiplanare Reformation (MPR) und ein Fenster enthält die Ansicht der 3D Segmentierungen. Die andere Ansichtsmöglichkeit ist für die Verwendung von zwei Monitoren ausgelegt und enthält zusätzlich noch eine Ansicht für eine 3D Darstellung der tomographischen Bilder. Das Laden der Bilder ist über einen *Patienten Browser* möglich, welcher auf eine lokal angelegte Datenbank zugreifen kann oder, wenn dieser mit dem Server verbunden ist, auf Daten dieses zugreifen kann. Die Bearbeitung ist in *Contouring Tools*, *Volumetric Tools*, *Drawing Tools*, *Object Operations* und *Object Combinations* eingeteilt. Unter der Kategorie *Contouring Tools* stehen dem Nutzer die Methode *Live Wire*, *Spline Curve* und *Polygone* zur Verfügung. Bei den *Volumetric Tools* stehen Segmentierungsmethoden über das *Thresholding* der Grauwerte sowie das *Region Growing* zur Verfügung. Die Segmentierungen werden im *Object Manager* verwaltet. Über diesen können die Farben der Segmentierungen geändert und die Segmentierungen aktiviert oder deaktiviert werden. Die Segmentierungen werden als Masken zwischengespeichert und können, bevor sie als Mesh generiert werden, noch bearbeitet werden. Zur Bearbeitung stehen Methoden wie Glättung, morphologische Operatoren oder einige andere Methoden zur Verfügung, die hier aber nicht alle genannt werden. Die erstellten Meshes können als Mesh im STL-Format oder als Maske als DICOM-Dateien abgespeichert werden. Es ist möglich, erstellte Meshes oder Masken in den Prototypen zu laden, welche dann im *Object Manager* aufgelistet werden. In Abbildung 4.2 ist ein Screenshot des Prototypen ohne die in der Arbeit entwickelten Erweiterungen zu sehen.

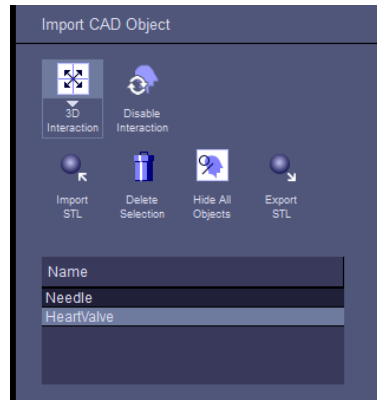


Abbildung 4.3: Importieren von Objekten

4.2.2 Erweiterungen

Der im vorherigen Abschnitt erklärte Prototyp wird um einige Funktionen erweitert, um die Ziele aus 1.2 umzusetzen. Die Erweiterungen werden in den folgenden Unterabschnitten dargestellt. Im Prototypen sind zwei Menüpunkte hinzugekommen, deren Funktion in den Kapiteln 4.2.2(Interaktion mit synthetischen Objekten) und 4.2.2(Smart Visibility) dargestellt wird, und die 3D Ansicht der Segmentierungen im unteren rechten Fenster des Prototypen wurde um eine zusätzliche Ansicht des CR mit Meshes erweitert.

Laden und Exportieren der synthetischen Objekte

Das Laden von synthetischen Objekten in den Prototypen wurde im Rahmen der Bachelorarbeit erweitert. Es erfolgt dynamisch über ein Pythonscript, sodass mehrere Objekte geladen werden können, welche dann in einer Liste verwaltet werden. Über die Liste ist es möglich die Namen der Objekte zu ändern, oder durch Auswählen eines Objekts die Interaktion mit diesem freizuschalten, es zu exportieren oder das Objekt zu löschen. Das Exportieren des ausgewählten Objektes ist im STL-Format möglich. In Abbildung 4.3 ist die grafische Oberfläche für das Laden und Exportieren der synthetischen Objekte zu sehen.

Interaktion mit synthetischen Objekten

Im Folgenden werden zuerst die Interaktionsmöglichkeiten in 3D beschrieben und dann für 2D.

Für die Umsetzung der Interaktion in dieser Arbeit wird sich an einem verfügbaren 3D Widget bedient, welches eine *Axis-Aligned Bounding Box* um das Objekt zeichnet. Dieses wird von MeVisLab bereitgestellt und ist in Abbildung 4.4 dargestellt. Das Modul heißt *SoTransformDragger* und lässt in diesem Beispiel die Interaktion mit einer Kugel (*SoSphere*) zu. In Abbildung 4.5 sind die verschiedenen Transformationen zu sehen.

Die Interaktion mit den Oberflächenschnitten der Objekte in 2D besitzt zwei Freiheitsgrade, wodurch sie zum Teil einfacher umzusetzen ist. Für die Translation ist kein zusätzliches *Widget* nötig, sondern ist eine Art *Drag and Drop*-Methode. Dafür muss die Maus in der Szene platziert werden, dann die linke Maustaste gedrückt werden und so lange gehalten werden bis das Objekt an der gewünschte Stelle platziert ist. Dies ist

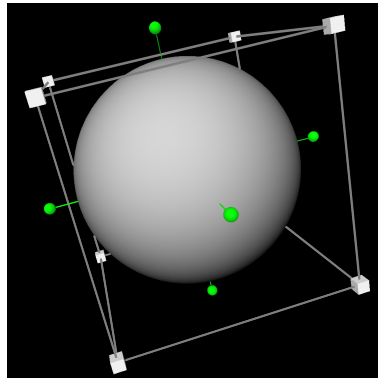
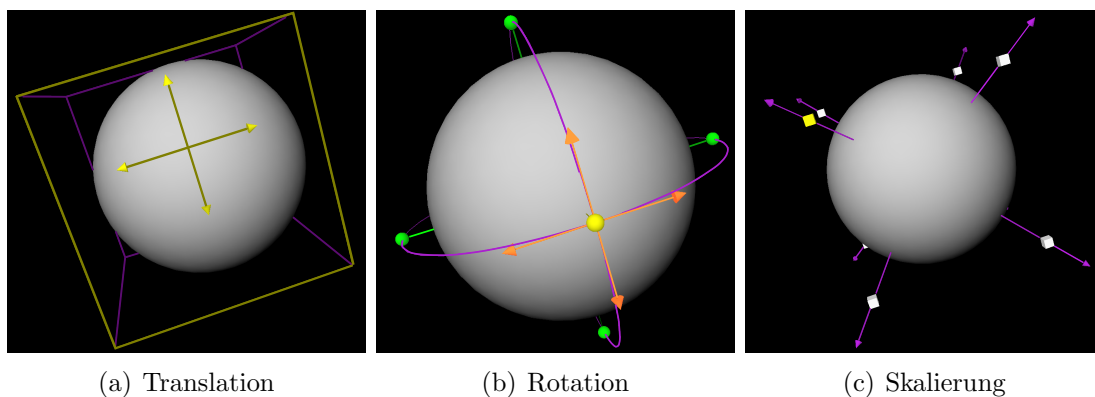


Abbildung 4.4: MeVisLab Modul *SoTransformDragger* mit einer Kugel (*SoSphere*)



(a) Translation

(b) Rotation

(c) Skalierung

Abbildung 4.5: Transformationen mithilfe von *SoTransformDragger*

eine für den Mensch aus vielen Bereichen bekannte Interaktionstechnik. Die Rotation und Skalierung wird, wie auch die 3D Interaktion, mithilfe von jeweils einem *Widget* vereinfacht, welche auch in MeVisLab vorhanden sind. Bei der Rotation wird ein Kreis um das Objekt gelegt, wobei der Mittelpunkt das Zentrum der *Bounding Box* ist. Der Kreis muss dann ausgewählt und festgehalten werden, bis das Objekt gewünscht rotiert ist. Für die Skalierung wird ein Rechteck, welches die Größe der *Bounding Box* hat, um das Objekt gezeichnet. Dieses kann an den Ecken ausgewählt und vergrößert werden. Alle drei Transformationen sind in Abbildung 4.6 abgebildet.

Smart Visibility kombiniert mit CR

Die Implementierung der *Smart Visibility* Methode ist in die vier Bereiche unterteilt:

- Schnittebene automatisch Platzieren über die PCA
- Veränderung der Form
- Erstellung der *Clipping Textur*
- Verarbeitung der *Clipping Textur* im CR

Die Implementierung der automatischen Platzierung der Schnittebene ist durch ein Modul realisiert, welches die PCA über die Knoten eines Meshes ausführt und die drei

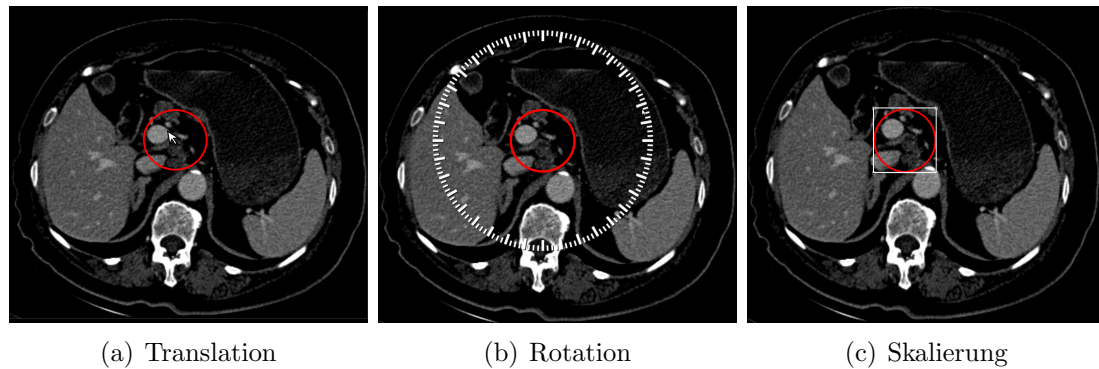


Abbildung 4.6: Transformationen für Objekte in 2D

Eigenvektoren ausgibt. Die Eigenvektoren werden verwendet, um die Schnittebene zu platzieren. Die Schnittebene kann über einen Button in der grafischen Oberfläche aktiviert werden. Durch das Auswählen einer Segmentierung im *Object Manager* wird die Schnittebene in der Ebene, die die zwei ersten Eigenvektoren der PCA der Segmentierung aufspannen, platziert.

Ein Beispielnetzwerk der Realisierung der letzten drei Punkte der Auflistung (Veränderung der Form, Erstellung der *Clipping Textur* und der Verarbeitung der *Clipping Textur* im CR) ist in Abbildung 4.7 zu sehen. Die ersten zwei Punkte sind durch je ein Makromodul realisiert. Die Veränderung der Form entweder die Vergrößerung des Meshes oder die Änderung der Form zu einem Primitiv wie Kugel, Würfel, Zylinder oder Kegel sind intern im Makromodul *CalculateShape* realisiert. Je nach Eingabe des Nutzers wird die Form geändert und vergrößert sowie die Größe des Meshes angepasst. Die Änderung der Form ist durch die in MeVisLab enthaltenen Inventormodule für die einzelnen Primitive umgesetzt. Die Änderung der Größe wird durch ein Pythonskript und das Inventormodul *SoScale* angepasst.

Die Erstellung der *Clipping Textur* ist in einem eigenen Makromodul *CinematicClip-WithDepth* realisiert. In diesem wird die gegebene Inventorszene als Textur umgewandelt und diese mit einem Shader, so wie in Kapitel 3.2.2 (Erstellung der Clipping Textur) beschrieben wurde, erzeugt. Das Weichzeichnen des Randes ist in einem eigenen Shader im Makromodul implementiert. Die Ausgabe dieses Makromoduls ist die *Clipping Textur*, die dann an den CR weitergegeben wird.

Die Verarbeitung der *Clipping Textur* wird im schon bestehenden Shader des CRs, so wie in Kapitel 3.2.2 (Verarbeitung der Textur im Cinematic Renderer) beschrieben, erweitert. Dieser Shader wird durch das Modul *SoCinematicRenderer* geladen. Die *SoCinematicTexture* Module sind für das Darstellen der zwei Transferfunktionen zuständig. Die Transferfunktionen können durch Mausinteraktionen, welche durch die Module *SoMouseGrabber* erfasst werden, in ihrer Fensterung verändert werden. In Abbildung 4.8 ist die grafische Oberfläche für das Ausschneiden der *Clipping Textur* zu sehen. Der Nutzer kann dies über den Button *Clip With Depth* aktivieren und dann auswählen, welches Objekt ausgeschnitten werden soll, entweder die Segmentierung (Mesh) oder die Objekte (CAD). Zusätzlich kann die Form über einen Button geändert werden und die Größe sowie das Weichzeichnen des Randes vergrößert werden.

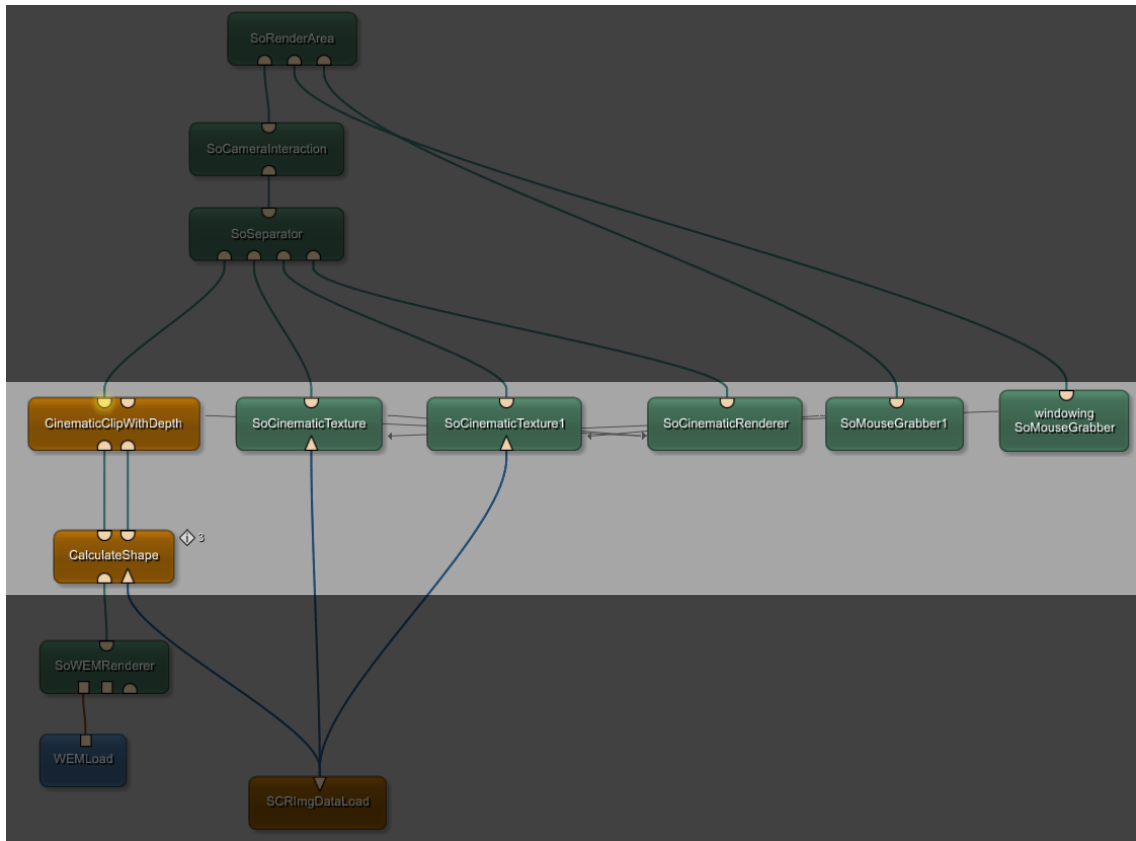


Abbildung 4.7: Netzwerk zur Smart Visibility

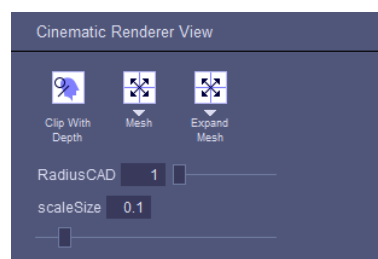


Abbildung 4.8: Smart Visibility im Prototyp

Umsetzung der Beleuchtung

Die Implementierung der Beleuchtung umfasst, das Einlesen der Beleuchtungsmaps, das Abtasten der Beleuchtungsmap und das Platzieren der Lichter in der 3D Szene. Jeder dieser Punkte wurde durch unterschiedliche Module entwickelt und alle zusammen in einem Macromodul vereint. Das interne Netzwerk dieses Makromoduls ist in Abbildung 4.10 zu sehen.

Das Einlesen der Beleuchtungsmap wird mit dem ML Modul *loadHDRImage* in C++ umgesetzt. Dieses liest die eingegebene HDR Datei aus und gibt ein *ML image* zurück, welches in MeVisLab weiterverarbeitet werden kann.

Das Abtasten der Beleuchtungsmap ist in dem Makromodul *ReadCubemap* und mit einem Pythonskript umgesetzt. Das Modul erhält das eingelesene HDR Bild und gibt eine Liste von Positionen mit Farbwerten zurück. Intern enthält das Makromodul das *XMarkerListContainer* Modul, in welchem die abgetasteten Punkte zur Weiterverarbeitung gespeichert werden. Dabei werden die Farbwerte und die Positionen der Punkte gespeichert.

Die Platzierung der Lichter in der Szene sollte zuerst durch das dynamische Erzeugen von *SoDirectionalLight* Modulen umgesetzt werden, da dieses Modul von MeVisLab bereitgestellt wird. Allerdings unterstützt der *OpenGL* Treiber des verwendeten Rechners die Verwendung von maximal acht *SoDirectionalLight* Modulen, weil OpenGL normalerweise nur acht Lichter (GL_LIGHT0 bis GL_LIGHT7) unterstützt [41]. Da beim Abtasten der Beleuchtungsmap mit einem Gitter 3.3.1 mehr als acht Lichtquellen entstehen können, ist die Einschränkung auf acht Lichter nicht akzeptabel. Deshalb wird die Erzeugung der Lichter in einem Shader implementiert, welcher im Inventormodul *SoFragmentShader* implementiert ist. Die Grundidee für den Shader stammt aus dem Tutorial [9] und wird für den Anwendungszweck leicht verändert. Ein Pseudocode des implementierten Shaders ist in 1 dargestellt. Der implementierte Shader erhält als Eingabe zwei Arrays, die jeweils Positionen und RGB-Werten der Lichtquellen enthalten, diese zwei Arrays sind im Pseudocode unter *Lights* abgespeichert. Es sind zwei Arrays, da für die Beleuchtung des CRs die *irradiance* und die *reflection* Beleuchtungsmap verwendet werden und diese für die Beleuchtung der synthetischen Objekte beide abgetastet werden. Als weitere Eingabe erhält der Shader Variablen mit Informationen über das Glänzen, die Intensität der Beleuchtung, welche manuell eingestellt werden kann und die Position des aktuell betrachteten Knotens (*vec*) auf der Oberfläche des synthetischen Objekts mit seiner Normale (*vecN*). Der Algorithmus in 1 wird für jeden Pixel des Resultatbildes ausgeführt. Die *For-Schleife* durchläuft jede Lichtquelle, da jede Lichtquelle auf jeden Pixel des Resultatbildes Einfluss hat. In der *For-Schleife* wird die Richtung des Vektors der aktuell betrachteten Lichtquelle zum aktuellen Knoten (*vec*) berechnet, dann der aktuelle Knoten in „Augenkoordinaten“ transformiert und die Richtung der Reflektion von Lichtquelle zu Knoten und Oberflächennormale des aktuellen Knotens berechnet.

Dann wird für den diffusen Term die Farbe nach dem Lambert Gesetz (siehe Kapitel 2.7.1) berechnet. In Abbildung 4.9(a) wird das Lambert Gesetz schematisch dargestellt. Für den diffusen Term wird der Winkel zwischen der Normalen und der Richtung von Knoten zur Lichtquelle gebildet und dieser Winkel mit der Farbe der aktuell betrachteten Lichtquelle multipliziert. Falls das Ergebnis kleiner Null oder größer Eins ist, wird dieses noch auf Null oder Eins gesetzt, da für die Farbe nur Werte zwischen Null und Eins zulässig sind.

Dann wird für jede Lichtquelle noch der spekularen Term berechnet. Für diesen wird der Winkel zwischen der schon berechneten Reflektion (R) und dem Vektor von Knoten zu Betrachter (E) berechnet und potenziert mit dem Faktor für das Glänzen. Die Potenz wird dann multipliziert mit der Farbe der *reflection* Lichtquelle. Eine grafische Darstellung der spekularen Reflektion ist in Abbildung 4.9(b) zu sehen. Das Ergebnis für die spekulare Reflektion wird ebenso wie für die diffuse Reflektion auf Null oder Eins gesetzt, falls es kleiner Null oder größer Eins wird. Am Ende der *For-Schleife* wird dann die Summe der diffusen Reflektion (*BeleuchtungIrr*) und der spekularen Reflektion (*BeleuchtungRef*) berechnet und zur *finaleFarbe* aufsummiert.

Nachdem alle Lichtquellen durchlaufen wurden, wird die *finaleFarbe* mit der Farbe des zu beleuchtendem Objekts (*farbeObjekt*) multipliziert. Das Ergebnis des Shaders ist die Farbe für jeden Pixel des resultierenden Bildes *FarbeProPixel*, also ein 2D Bild der Szene.

Algorithm 1 Beleuchtung pro Pixel

```

procedure BELEUCHTUNG PRO PIXEL (Lichter, Intensität, Glanz, vec,
vecN)
  N = normieren(vecN)
  initialisiere Variable finaleFarbe
  for i = 0 TO Lichter.count do
    LichtPosition = Lichter.position
    FarbeIrr = Lichter.irradiance
    FarbeRef = Lichter.reflection

    L = normieren(lightPos - vec)
    E = normieren(-vec) // Transformation in Augenkoordinaten
    R = normieren(-reflect(L,N)) // Richtung der Reflektion von
    Lichtstrahl(L) zur Oberflächennormalen

    // Diffuser Term
    BeleuchtungIrr = FarbeIrr * max(dot(N,L),0.0)
    BeleuchtungIrr = clamp(BeleuchtungIrr*Intensität,0.0,1.0)

    // Spekularer Term
    BeleuchtungRef = FarbeRef * pow(max(dot(R,E),0.0),Glanz*0.3)
    BeleuchtungRef = clamp(BeleuchtungRef*Intensität,0.0,1.0)

    finaleFarbe += BeleuchtungIrr+BeleuchtungRef
  end for
  // Farbe pro Pixel
  FarbeProPixel = finaleFarbe * farbeObjekt
end procedure

```

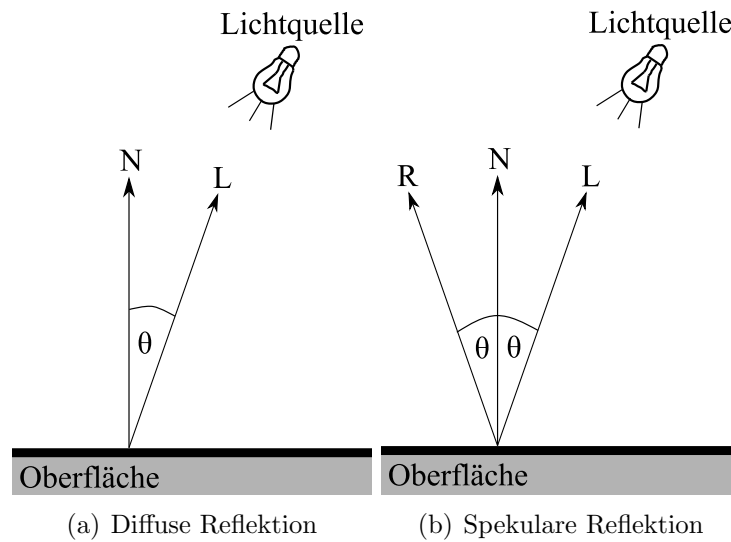


Abbildung 4.9: Diffuse und Spekulare Reflektion (nach [12])

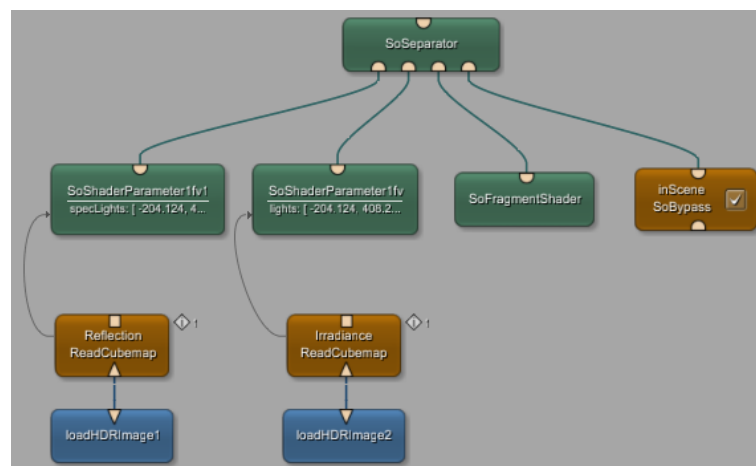


Abbildung 4.10: MeVisLab Netzwerk zur Beleuchtung

Labeling

Das Platzieren der Labels in 2D ist bereits für MeVisLab in einem Modul implementiert gewesen, weshalb darauf im folgenden nicht mehr eingegangen wird. Die Platzierung in 3D wird mit einem Inventormodul in C++ realisiert. Dieses erhält als Eingabe eine Liste mit Informationen über die zu labelnden Objekte, welche den Namen der Objekte, dem Volumen der Objekte und der Position enthält. Im C++ Code wird dann so wie im Kapitel 3.1 bereits beschrieben wurde die Platzierung der Labels berechnet und mit *OpenGL* Befehlen die Verbindungslinie und das Label gezeichnet. Dies ist auch die Ausgabe des Moduls.

Kapitel 5

Evaluierung

In diesem Kapitel wird die Evaluierung der verbesserten Visualisierung durchgeführt. Die Erweiterungen, die entwickelt wurden und zu einer verbesserten Visualisierung beitragen, sind quantitativ nicht messbar, weshalb auf eine quantitative Evaluierung verzichtet wird. Zusätzlich muss gesagt werden, dass aufgrund der eingeschränkten Bearbeitungszeit, die für die Bachelorarbeit vorgesehen ist, keine umfangreiche Evaluierung durchgeführt werden kann. Dennoch wurde ein kleiner Nutzertest durchgeführt. Eine weitere Einschränkung besteht darin, dass keine Nutzer (Ärzte) befragt werden können, für die das Produkt entwickelt wurde. Da ein Ziel allerdings ist, dass die Arzt-Patienten-Kommunikation verbessert werden soll, ist es auch wichtig nicht Mediziner zu den Visualisierungen zu fragen.

5.1 Versuchsaufbau und Durchführung

Der Versuchsaufbau umfasst die Erstellung von drei Fragebögen, die im Folgenden dargestellt werden und die Bereitstellung eines Arbeitsplatzes mit dem installierten Prototypen. Die Durchführung des Nutzertest wird im nachfolgenden Kapitel beschrieben.

5.1.1 Fragebögen

Für die Evaluierung werden drei selbst erstellte Fragebögen verwendet. Diese sind unterteilt in die drei Schwerpunkte der Arbeit, einmal die Smart Visibility kombiniert mit dem CR, die Interaktion mit den synthetischen Objekten und die Beleuchtung der synthetischen Objekte. Die Fragebögen sind im Anhang A beigefügt und werden in den nächsten Unterabschnitten kurz erläutert. Alle außer zwei Fragen erlauben Antworten auf einer *5-Point-Likert-Skala*¹

Smart Visibility kombiniert mit Cinematic Rendering

Der Fragebogen über die Smart Visibility ist unterteilt in zwei Kategorien, einmal Fragen zur Schnittebene und einmal in Fragen zur neuen Visualisierungsmethode, dem Ausschneiden der *Clipping Textur*. Die zwei Kategorien beinhalten jeweils die selben drei

¹Eine Likert Skala ist eine Methode für Skalierung von Antworten. Die Skala ist bipolar und mit Eigenschaften verknüpft, denen die Befragten mehr oder weniger stark zustimmen[28].

Fragen. Die drei Fragen beziehen sich auf die Darstellung von Informationen eines synthetischen Objekts, welches im Nutzertest Segmentierungen oder geladene CAD-Modelle sind. Mit Informationen sind Tiefeninformationen der 3D Szene, die Lage des Objekts und die Größe des Objekts gemeint. Zusätzlich wird gefragt, wie schnell die Informationen erfasst werden können und, ob die Visualisierung für Patientenaufklärung vor Therapien verwendet werden kann. Am Ende des Fragebogens ist noch der Punkt „Allgemeine Anmerkungen“, unter diesem können andere nicht zu den Fragen passende Anmerkungen aufgefasst werden.

Interaktion

Der Fragebogen zu den Interaktionsmöglichkeiten mit den synthetischen Objekten beinhaltet zwei Abschnitte. Zum einen Fragen zur Interaktion in 2D und zur Interaktion in 3D. Dabei werden in jedem Abschnitt dieselben Fragen gestellt. Mit der ersten Frage soll herausgefunden werden, ob die jeweilige Art der Transformation (Skalierung, Rotation und Translation) des Objekts intuitiv ist. Die zweite Frage ist, ob die Interaktion zielführend ist. Zusätzlich zu den zwei Fragen und zwei Abschnitten können am Ende des Fragebogens noch „allgemeine Anmerkungen“ zur Interaktion gemacht werden.

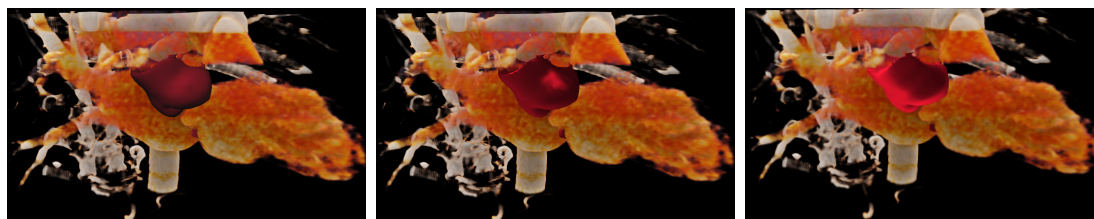
Beleuchtung

Der Fragebogen zur Beleuchtung beinhaltet drei Fragen, mit denen zweimal drei Bilder verglichen werden sollen. Also die Bilder 1.-3. und die Bilder 4.- 6. sollen verglichen werden. Die erste Frage soll herausfinden, wie homogen die Darstellung des Objektes zum Volumen passt. Dabei ist das Objekt in den zwei Bilderreihen jeweils eine Segmentierung. Die zweite und dritte Frage sind identisch und sollen die Bilder 1.- 3. und 4.-6. vergleichen. Mit den Fragen soll evaluiert werden, welches Bild sich am besten zur Patientenaufklärung vor einer Therapie eignet. Als Antwort ist nur eines der jeweils drei Bilder möglich. Auch bei diesem Fragebogen können noch „allgemeine Anmerkungen“ zur Beleuchtung gemacht werden.

5.1.2 Ablauf

Für die Evaluierung wurde ein Nutzertest mit fünf Probanden und Probandinnen² durchgeführt. Die Probanden waren Studenten der Medizin oder *Clinical Coaches*. Dabei haben die Probanden den Prototypen mit den neuen 3D Visualisierungen mit einem CT Datensatz eines Herzens und einem CT Datensatz einer Leber verwendet. Alles was die Probanden während der Benutzung des Prototypen gesagt haben, wurde mitgeschrieben und ist im Kapitel 5.2 mit aufgeführt. Nach einer kurzen Einführung, welche die Motivation der Arbeit, die Erklärung des Prototypen und eine Erinnerung daran, dass sie alle ihre Gedanken zum Prototypen laut aussprechen sollen, beinhaltet hat, wurde der erste Datensatz geladen, welches die CT Aufnahme eines Herzens ist und eine Segmentierung einer Aortenklappe beinhaltet. Dann wurden die Probanden aufgefordert sich einen Überblick über 3D Visualisierung mit Fokus auf die segmentierte Herzklappe im Volumen

²Im weiteren Verlauf wird nur noch die männliche Form des Wortes *Proband* verwendet. Damit ist aber auch immer die weibliche Form gemeint.



(a) ohne Beleuchtung, Fragebogen 1.Bild (b) mit drei Lichtquellen, Fragebogen 2.Bild (c) mit gleichmäßig verteilten Lichtquellen, Fragebogen 3.Bild

Abbildung 5.1: Bilder für die Auswertung der Beleuchtung Datensatz mit Herz

zu machen. Als Hilfsmittel stand die in Abschnitt 3.2.1 präsentierte Schnittebene zur Verfügung und den Probanden wurde gezeigt, wie sie die Transferfunktion ändern sowie diese fenstern können. Nach ca. 5-10 Minuten haben sich die Probanden vertraut mit der 3D Visualisierung und der Schnittebene gefühlt. Dann wurde ihnen erklärt, wie sie ein Objekt in den Prototypen laden können und sie wurden aufgefordert die in 2.1.2 erwähnte Herzklappe zu laden. Diese sollten sie dann an die Stelle der segmentierten Herzklappe platzieren und auch die Skalierung und Rotation an die segmentierte Herzklappe angleichen, dafür standen ihnen die Interaktionen in 2D und in 3D zur Verfügung. Nachdem die Probanden die Herzklappe an eine für sie richtige Position platziert haben, wurde ihnen gezeigt, wie sie den Modus für eine Draufsicht auf das Modell der Herzklappe anstellen können. Diesen haben sie dann angestellt und ihnen wurde der Unterschied gezeigt, der zwischen dem Ausschneiden der unveränderten Form der Herzklappe oder dem Ersetzen der Form durch eine einfache Form (3.2.2) entsteht. Nun hatten sie einige Minuten Zeit die veränderte Visualisierung aus allen Kameraperspektiven zu betrachten, sowie die zweite Transferfunktion für das Volumen der *Clipping Textur* zu ändern und zu fenstern. Dann wurde der zweite Datensatz, der die Aufnahme einer Leber mit einer Segmentierung der Leber, den Blutgefäßen der Leber und einem Tumor in der Leber enthält, geladen. An diesem sollten sie die entwickelte Smart Visibility Methode ein zweites Mal testen. Dafür konnten sie die Form der unveränderten segmentierte Leber als Grundlage für die *Clipping Textur* verwenden oder die Leber durch eine einfachere Form (3.2.2) ersetzen. Durch das Ausschneiden der Leber werden die segmentierten Blutgefäße und der Tumor sichtbar. Auch für den zweiten Datensatz hatten sie einige Minuten Zeit die 3D Visualisierung zu betrachten und die Transferfunktion sowie die Fensterung zu ändern. Als nächstes wurde den Probanden der Fragebogen über die Smart Visibility und über die Interaktion gegeben. Nachdem sie diesen ausgefüllt hatten, wurde ihnen der letzte Teil der in der Arbeit entwickelten Erweiterungen gezeigt, die Beleuchtung der synthetischen Objekte. Diese wurde an Bildern, die in Abbildung 5.1 und 5.2 zu sehen sind, bewertet. Die Probanden wurden nicht informiert, in welchem Bild die Objekte wie beleuchtet sind. Sie hatten dann ca. fünf Minuten Zeit, die Bilder immer wieder zu betrachten, dann wurde den Probanden der Fragebogen über die Beleuchtung gegeben. Am Ende wurden sie noch gefragt, ob sie noch zusätzliches Feedback haben. Der gesamte Test hat im Durchschnitt 40 Minuten gedauert.

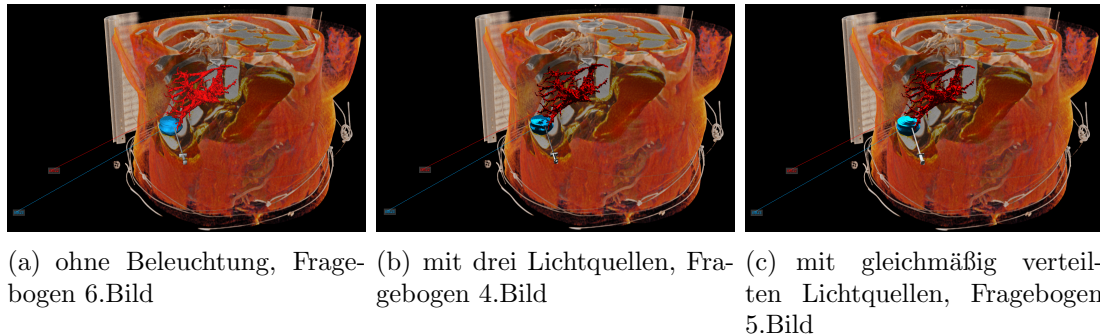


Abbildung 5.2: Bilder für die Auswertung der Beleuchtung Datensatz mit Leber

5.2 Ergebnisse und Auswertung

In diesem Kapitel werden die Ergebnisse des Nutzertest präsentiert und ausgewertet. Allgemein ist zu sagen, dass durch den Nutzertest nur Tendenzen festgestellt und grobe Fehler aufgedeckt werden können. Dies liegt auch daran, dass der Prototyp noch in einem recht frühen Stadium ist. Die grafische Oberfläche zum Beispiel sollte für einen kommerziellen Einsatz noch verbessert werden und die Performanz sollte auch noch verbessert werden. Es wird empfohlen einen Nutzertest nach Einbau der Verbesserungen in einem größerem Rahmen zu wiederholen sowie mit mehreren Probanden und Probanden die Endnutzer des Prototyps sind. Die genauen Ergebnisse der Fragebögen sind in den folgenden Kapiteln als Säulendiagramme dargestellt.

5.2.1 Ergebnisse der Smart Visibility kombiniert mit Cinematic Rendering

Im folgenden wird der Fragebogen zur *Smart Visibility* ausgewertet, die allgemeinen Aussagen der Probanden zum Thema präsentiert und die Ergebnisse ausgewertet. Dabei wird auf den Vergleich zwischen dem Ausschneiden von Teilen des Volumens mit Schnittebene oder mit *Clipping Textur* eingegangen. Die erste Frage ist: „Wie werden die jeweiligen 3D Informationen des Objekts visualisiert?“. Für die erste Kategorie der Informationen, die Tiefeninformationen, kam heraus, dass es für die Tiefeninformationen keinen Unterschied macht, ob die Schnitteben verwendet wird oder das Ausschneiden des Volumen zwischen Betrachter und Objekt. Die zweite Kategorie der Informationen behandelt die Lage des Objekts. Dabei kam heraus, dass die Lage etwas besser bewertet werden kann, wenn das Volumen zwischen Betrachter und Objekt ausgeschnitten wird, als wenn nur die Schnittebene zur Verfügung steht. Für die dritte Kategorie, die Größe des Objekts, empfanden die Probanden keinen Unterschied bei der Wahrnehmung der Information, zwischen Verwenden der Schnittebene oder dem Ausschneiden des Volumen zwischen Betrachter und Objekt. Die zweite Frage ist: „Wie schnell können die dargestellten Informationen erfasst werden?“. Hierbei kam heraus, dass die Probanden die Informationen etwas schneller durch das Ausschneiden des Volumen zwischen Betrachter und Objekt erfassen konnten. Die letzte Frage des Fragebogens ist: „Wie nützlich sind die Visualisierungen für die Patientenaufklärung vor Therapien??“. In dieser Frage gab es keinen Unterschied bei der Beantwortung der Frage zwischen Schnittebene und

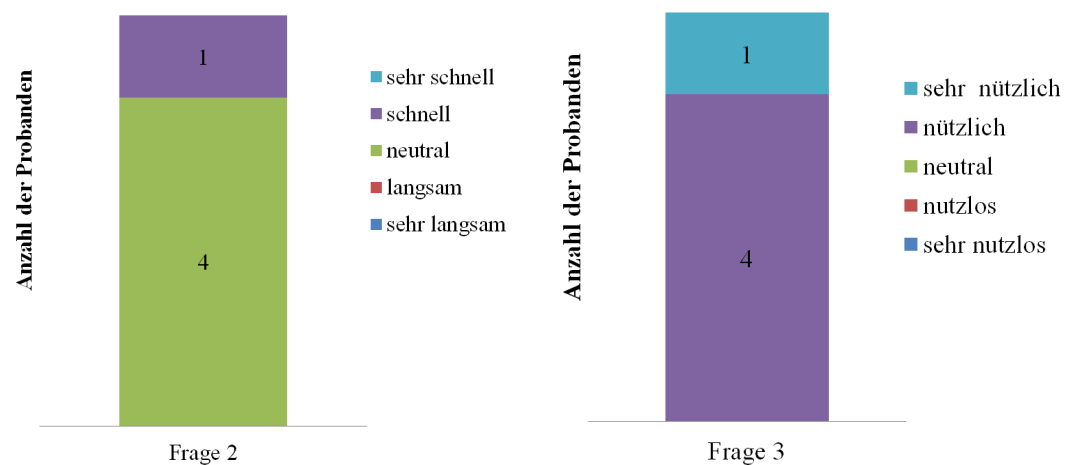
Clipping Textur. Die Probanden empfinden beide Techniken als nützlich bis sehr nützlich zur Patientenaufklärung vor Therapien.

Eine allgemeine Aussage war, dass für die Patientenaufklärung besser fertig gerenderte Bilder der 3D Visualisierung gezeigt werden sollten, da der CR das gerendert Volumen bei jeder Bewegung der Kamera oder eines Objekts in der Szene neu iterativ rendert. Dadurch gibt es Momente, in denen die 3D Szene sehr verpixelt erscheint. Dies könnte den Patienten verwirren. Eine andere Aussage war, dass gerade für die Patientenaufklärung mehr 3D Visualisierungen gewünscht sind, da diese für nicht Mediziner viel einfacher zu verstehen sind als die Schichtbilder. Eine weitere interessante nur zum Ausschneiden des Objekts mit *Clipping Textur* war, dass die Informationen verstanden werden, solange der Proband die Kamera rotiert, aber so bald die Kamera losgelassen wird und das Bild statisch ist, sind die Informationen nicht mehr gut zu begreifen. Ein Proband hat angemerkt, dass bei Verwendung der *Clipping Textur* die Tiefeninformationen vielleicht besser erfasst werden könnten, wenn das Objekt Schatten in das Volumen wirft oder das Volumen Schatten auf das Objekt wirft.

Das Ausschneiden mit *Clipping Textur* kann für die Erkennung der Lage eines Objekts nützlich sein, natürlich nur, wenn durch die *Clipping Textur* das Volumen so ausgeschnitten wird, dass das Objekt sichtbar wird. Die Tiefeninformationen können beim Ausschneiden des Volumen mit *Clipping Textur* nicht besser wahrgenommen werden als mit der Schnittebene. Die Aussage eines Probanden war, dass Schatten dabei helfen könnte die Tiefeninformationen wahrzunehmen. Dies ist eine gute Überlegung und sollte in weiteren Entwicklungen bedacht werden. Ein Ergebnis, welches für diese Arbeit wichtig ist, ist das allgemein 3D Visualisierung für die Patientenaufklärung vermehrt benutzt werden sollten.



Abbildung 5.3: Diagramm zur Frage der Schnittebene: „Wie werden die jeweiligen 3D Informationen des Objekts visualisiert?“



(a) „Wie schnell können die dargestellten Informationen erfasst werden?“

(b) „Wie nützlich sind die Visualisierungen für die Patientenaufklärung vor Therapien?“

Abbildung 5.4: Ergebnisse zur 2. und 3. Frage der Schnittebene

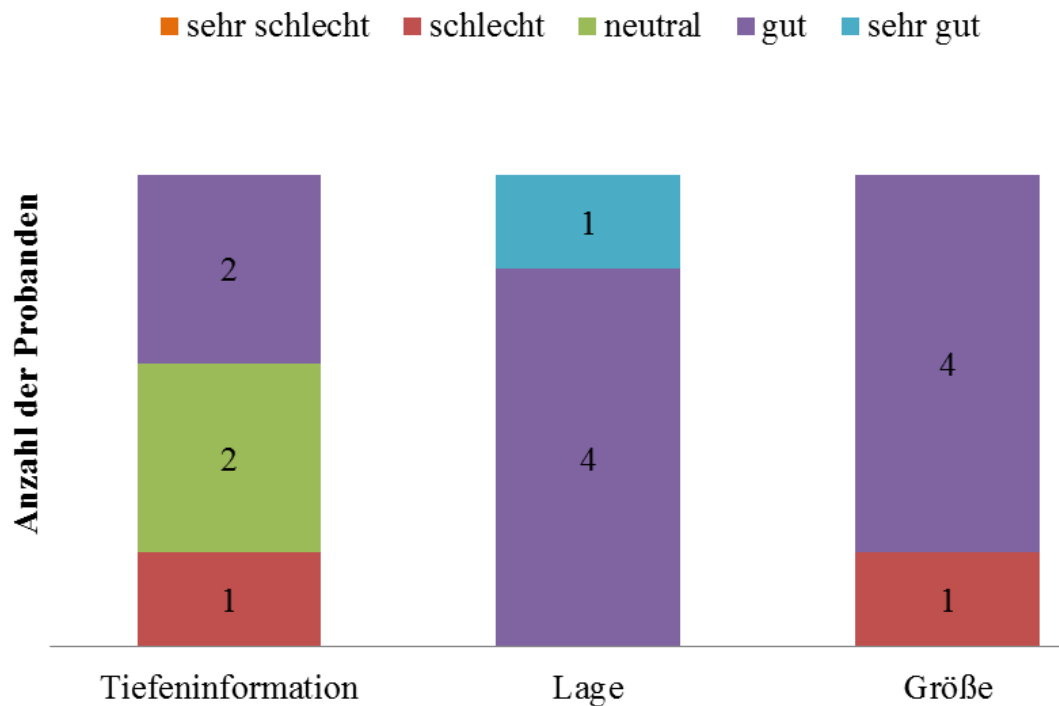
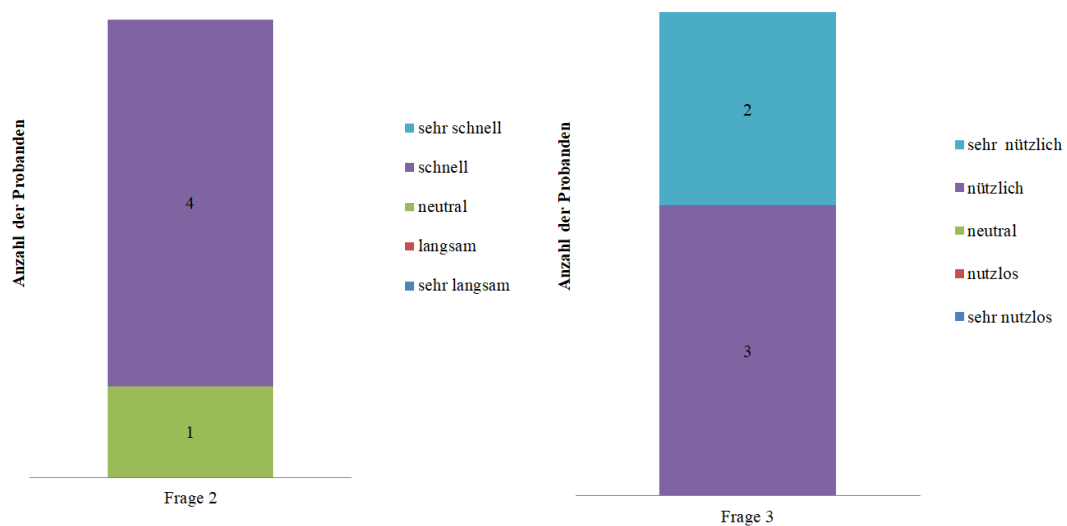


Abbildung 5.5: Ergebnisse zur Frage der *Clipping Textur*: „Wie werden die jeweiligen 3D Informationen des Objekts visualisiert?“



(a) „Wie schnell können die dargestellten Informationen erfasst werden?“

(b) „Wie nützlich sind die Visualisierungen für die Patientenaufklärung vor Therapien?“

Abbildung 5.6: Ergebnisse zur 2. und 3. Frage der *Clipping Textur*

5.2.2 Ergebnisse der Interaktion

Im folgenden wird der Fragebogen zur *Interaktion* ausgewertet, die allgemeinen Aussagen der Probanden zum Thema präsentiert und die Ergebnisse ausgewertet. In den nächsten Abschnitten werden zuerst die Ergebnisse zur 2D Interaktion präsentiert und ausgewertet und dann zur 3D Interaktion präsentiert und ausgewertet.

2D Interaktion

Die erste Frage war: „Wie intuitiv sind die einzelnen Interaktionen?“. Die Skalierung empfanden die Probanden von *neutral* bis *sehr intuitiv*, die Rotation empfanden sie als *intuitiv* bis *sehr intuitiv* und alle drei Probanden empfanden die Translation als *sehr intuitiv*. Die zweite Frage lautet: „Ist die Interaktion zielführend?“. Zwei Probanden empfanden die Interaktion als *zielführend* und ein Proband als *sehr zielführend*. Eine Allgemeine Aussage, die zur Interaktion gemacht wurde, war, dass die Skalierung immer nur in eine Dimension gemacht wird. Dies sei negativ anzumerken.

Dies ist wahrscheinlich auch der Grund, weshalb die Interaktion etwas schlechter als die Rotation oder Translation abgeschnitten hat. Allgemein war es erstaunlich, dass alle Probanden ohne Erklärung die Art der jeweiligen Interaktion erkannt haben und umsetzen konnten.

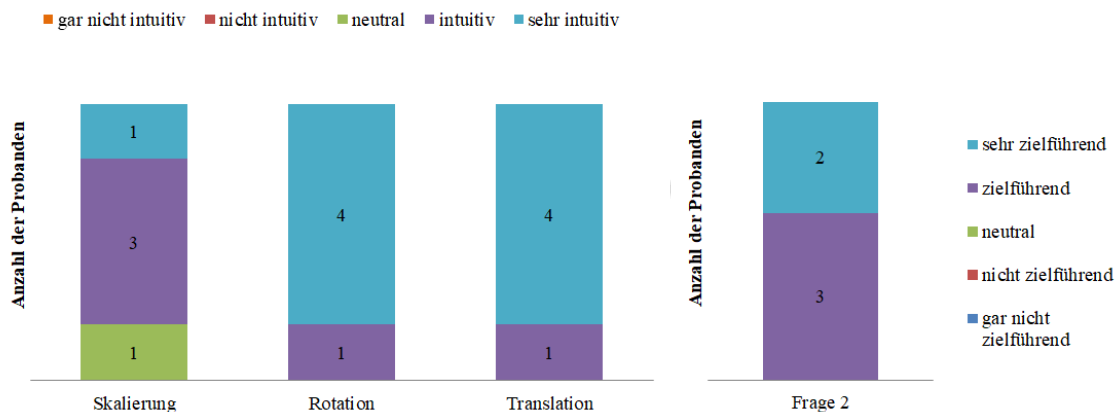


Abbildung 5.7: Ergebnisse zu den Fragen „Wie intuitiv sind die einzelnen Interaktionen?“ und „Ist die Interaktion zielführend?“ zur Interaktion in 2D

3D Interaktion

Zur Interaktion in 3D wurden dieselben Fragen, wie zur 2D Interaktion gestellt. Die Rotation hat dabei am besten abgeschnitten, die Skalierung wurde mit *neutral* bewertet und die Translation war für einen Probanden nicht intuitiv. Die Frage, ob die Interaktion zielführend ist, wurde etwas schlechter bewertet als die Interaktion in 2D. Eine Allgemeine Aussage war, dass die Interaktion in 3D erst nicht so einfach war und sich der Proband erst einmal an die Art gewöhnen muss. Eine anderer Proband hat ausgesagt,

dass die Lage des Objekts in der Szene schlecht erkannt werden kann und deswegen die Interaktion schwierig ist.

Dass die Interaktion in 3D nicht so gut abgeschnitten hat wie die Interaktion in 2D, kann damit begründet werden, dass das Medium, mit dem die Interaktion durchgeführt wird, eine Maus ist, die für 2D Interaktionen ausgelegt ist. Die Translation ist am schlechtesten von den drei Transformationen bewertet worden. Dies kann daran liegen, dass es bei der Translation drei Freiheitsgrade gibt. Das Bewegen vertikal, horizontal und lateral. Bei der Skalierung gibt es nur einen Freiheitsgrad und der Rotation zwei Freiheitsgrade, weshalb diese Interaktionen mit einer 2D Maus einfacher umzusetzen sind.

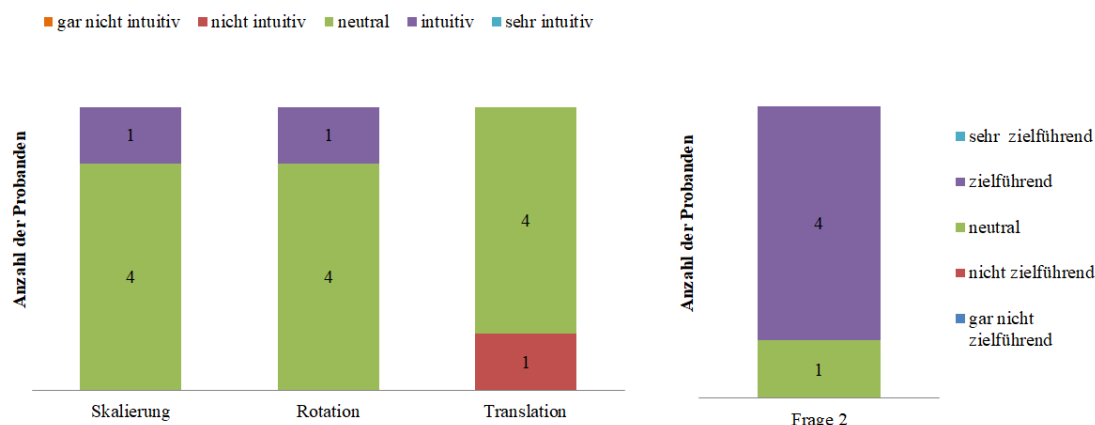


Abbildung 5.8: Ergebnisse zu den Fragen „Wie intuitiv sind die einzelnen Interaktionen?“ und „Ist die Interaktion zielführend?“ zur Interaktion in 3D

5.2.3 Ergebnisse der Beleuchtung

Am besten haben die Bilder abgeschnitten, bei denen die Objekte durch drei Lichtquellen beleuchtet werden. Die anderen zwei Kategorien, die Kategorie ohne Beleuchtung und mit Beleuchtung durch gleichmäßig verteilte Lichtquellen, haben sehr ähnlich abgeschnitten. Auf die Frage: „Mit welchem Bild kann man die Therapieplanung einem Patienten am besten zeigen?“ wurde unterschiedlich geantwortet. Bei dem Datensatz mit der Aufnahme des Herzens wurde zweimal für das 2.Bild (Beleuchtung durch drei Lichtquellen) gestimmt und einmal für das 3.Bild (Beleuchtung durch gleichmäßig verteilte Lichtquellen). Beim zweiten Datensatz mit der Aufnahme der Leber wurde zweimal für das 4.Bild (Beleuchtung durch drei Lichtquellen) und einmal für das 6.Bild (ohne Beleuchtung) gestimmt. Ein Proband hat die Anmerkung gemacht, dass der Schatten innerhalb des Volumens fehlt, dadurch kann die Tiefeninformation zwischen Volumen und Objekt nicht wahrgenommen werden.

Allgemein haben sich die Probanden schwer getan die Bilder mit der unterschiedlich beleuchteten Herzklappe (1.-3.) zu bewerten. Dies könnte daran liegen, dass hier nur ein synthetisches Objekt in der Szene ist und durch das Einschalten der Beleuchtung kein Schatten entsteht. Zwei der drei Probanden empfanden den Schatten, der im Datensatz mit Aufnahme der Leber auf den Tumor oder die Gefäße geworfen wurde, als nützlich, um die räumlichen Relationen besser zu verstehen. Wenn der Nutzertest noch einmal mit mehreren Probanden durchgeführt wird, könnte auch ein Video zur Bewertung der

Beleuchtung gezeigt werden. Dies könnte die Bewertung für die Probanden erleichtern, da die beleuchteten Objekte aus mehreren Blickwinkeln gezeigt werden könnten.

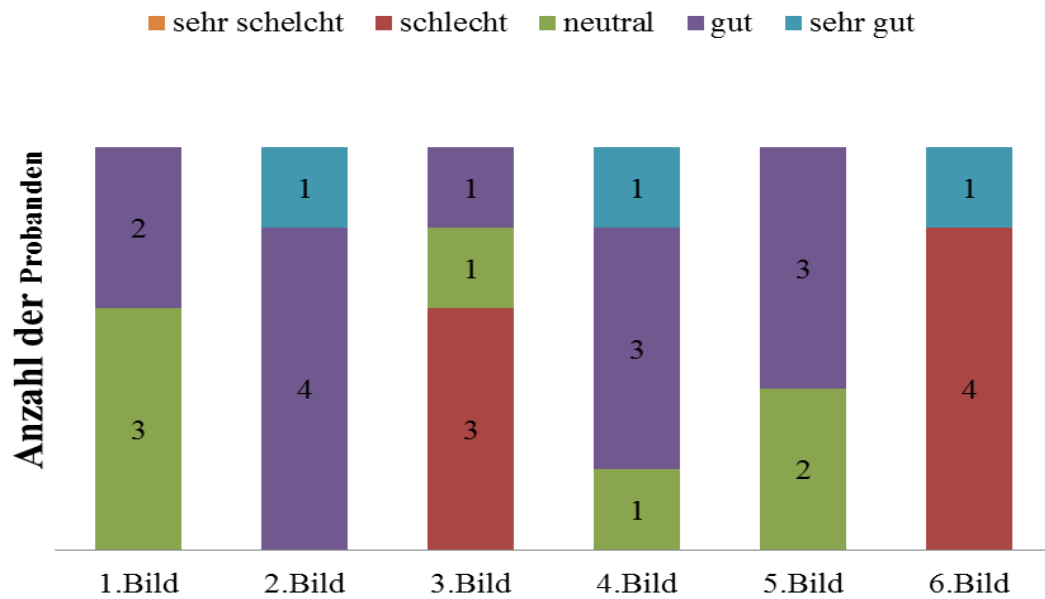


Abbildung 5.9: Ergebnisse zur Frage: „Wie homogen passt die Darstellung des Objekts zum Volumen?“

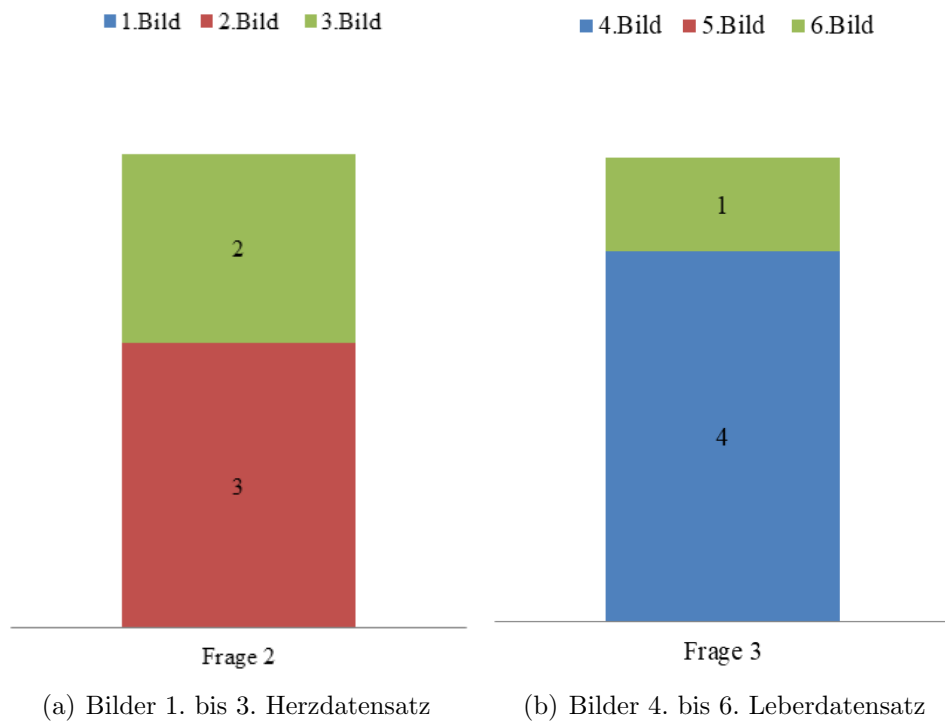


Abbildung 5.10: Ergebnisse zur Frage: „Mit welchem Bild kann man die Therapieplanung einem Patienten am besten zeigen?“

Kapitel 6

Zusammenfassung und Ausblick

In diesem Kapitel wird eine Zusammenfassung zu vorliegenden Arbeit sowie ein Ausblick auf mögliche Erweiterungen gegeben.

6.1 Zusammenfassung

Ziel dieser Arbeit war es, eine verbesserte Visualisierung für Therapieplanung mit dem CR zu entwickeln (siehe Kapitel 1.2) und die Erweiterungen beispielhaft an einem Prototypen zu zeigen.

Folgende Punkte wurden entwickelt:

- Ständige Sichtbarkeit auf synthetischen Objekte beim Rendern von tomographisch aufgenommen Daten mit dem CR
- Interaktive Manipulation von synthetischen Objekte und Beleuchtung dieser Objekte
- Semantisches Labeln von synthetischen Objekte in der gerenderten Szene
- Export von synthetischen Objekten in STL-Format für den 3D Druck

Der wichtigste Punkt und auch Hauptteil dieser Arbeit war das entwickeln einer *Smart Visibility* Methode, mit welcher die synthetischen Objekte dem Nutzer immer sichtbar sind, auch wenn sie eigentlich im CR gerendertem Volumen liegen würden. Zusätzlich sind sie aus allen Kamerapositionen sichtbar. Durch das Ändern der zweiten Transferfunktion oder durch Fenstern der zweiten Transferfunktion, kann das Volumen, welches ausgeschnitten wird, auch in einer anderen Farbe als das restliche Volumen dargestellt werden.

Die Interaktion mit den synthetischen Objekten ist in 2D und in 3D möglich. Die verschiedenen Transformationen können über die grafische Oberfläche des Prototypens ausgewählt werden. Die Beleuchtung der synthetischen Objekte wurde im Rahmen der Arbeit soweit umgesetzt, dass die Objekte beleuchtet werden und sie Schatten auf sich gegenseitig werfen. Dabei wird für die Beleuchtung, die Beleuchtungsmap die auch im CR verwendet wird, genutzt. Eine Erweiterung der Beleuchtung durch Schatten vom CR gerenderten Volumen auf die Objekte oder von den Objekten auf das Volumen ist

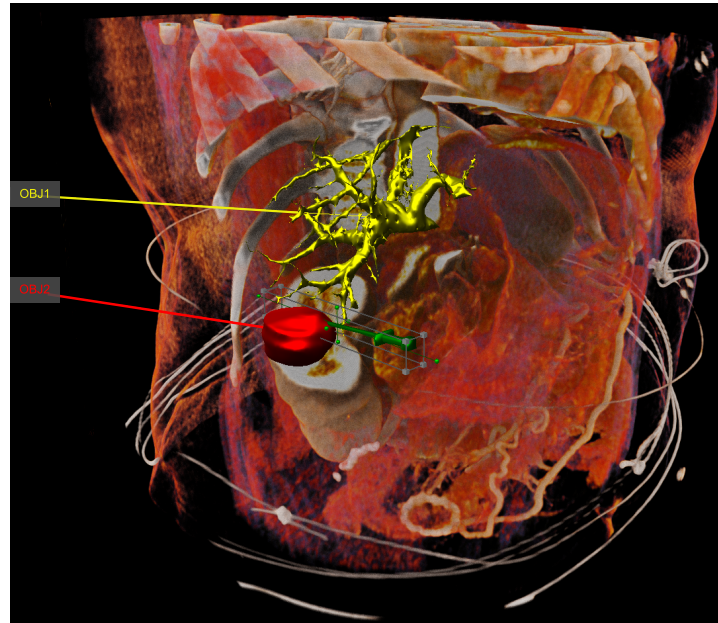


Abbildung 6.1: Endergebnis

denkbar und wird im Ausblick dargestellt.

Das Labeln von synthetischen Objekten wurde in 3D neu entwickelt. Die Labels werden außerhalb des Volumens mit einer Verbindungslinie zum Ankerpunkt des Objekts platziert. So sind die Labels sichtbar und werden nicht durch Teile des Volumens platziert. Das Exportieren der synthetischen Objekte in STL-Format wurde hinzugefügt, sodass nach Manipulation der Objekte diese für weiter Verarbeitungen genutzt werden können oder mit einem 3D Drucker ausgedruckt werden können.

Zur Evaluierung der Erweiterungen wurde ein Nutzertest durchgeführt, bei dem Probanden den Prototypen ausprobieren und die Visualisierungen bewerten sollten. Der Nutzertest hat ergeben, dass die 3D Visualisierungen für die Patientenaufklärung und für einen schnellen Überblick gewünscht sind. Damit sie auch für die Therapieplanung eingesetzt werden können, muss die Tiefeninformation besser dargestellt werden und der Cinematic Renderer schneller das Volumen rendern. Beide Interaktionsarten (2D und 3D) wurden als nützlich bewertet. Auch, wenn die Interaktion in 3D erst etwas ungewohnt erscheinen mag, haben sich die Probanden nach kurzer Eingewöhnungszeit mit der Art der Interaktion vertraut gefühlt.

Die Ziele, die in Kapitel 1.2 aufgestellt wurden, sind somit erfolgreich umgesetzt wurden. Alle Erweiterungen wurden in den in Kapitel 4.2 vorgestellten Prototypen eingebaut und können über die grafische Oberfläche der Prototyps bedient werden. Ein Visualisierung mit allen entwickelten Erweiterungen ist in Abbildung 6.1 und 6.2 zu sehen. Abbildung 6.1 zeigt den Leberdatensatz, welcher auch in der Evaluierung verwendet wurde. Dabei wird der Teil des Volumens transparent dargestellt, an dem die Leber positioniert ist. Dadurch werden die Blutgefäße (gelb) und ein Tumor (rot), welcher in der Leber liegt, sichtbar. Mit der Nadel kann die Tumorablektion geplant werden. Abbildung 6.2 zeigt den Herzdatensatz mit dem Aortenklappenmodell, welches in 2.1.2(Herzklappendefekt) vorgestellt wurde, und einer Segmentierten Aortenklappe (rot).

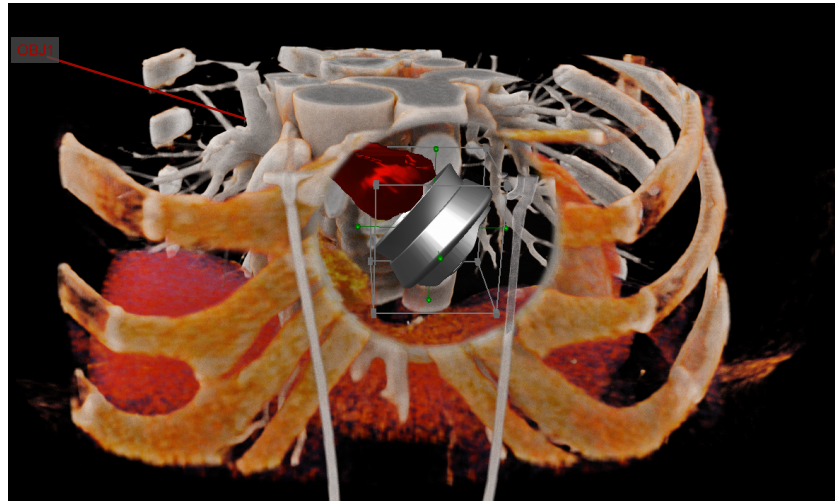


Abbildung 6.2: Endergebnis Herzdatensatz

6.2 Ausblick

In diesem Kapitel werden mögliche Anwendungen und Erweiterungen für die in der Arbeit entwickelte Visualisierung beschrieben.

Der aktuelle Stand des Prototyps wurde zwei *Clinical Coaches* gezeigt, um mögliche Anwendungsbereiche für die entwickelte Visualisierung zu finden. Dabei wurden folgende Anwendung für Therapieplanung oder Diagnosezwecke genannt.

6.2.1 Weitere Anwendungen

Die Anwendungsgebiete lassen sich in Anwendungen für die Therapieplanung und Diagnose unterteilen. Diese werden in den folgenden Kapiteln kurz dargestellt. Folgende Erkrankungen können mit der in der Arbeit vorgestellten Visualisierung diagnostiziert oder die Therapie für diese Erkrankung geplant werden:

- Schlaganfall
- Traumatologie
- Leberresektion
- Therapie von Aneurysmen

Bei einem Schlaganfall wird ein Bereich im Gehirn nicht mehr mit Blut versorgt, wodurch die Nervenzellen nicht mehr genug Sauerstoff erhalten und absterben [42]. Je schneller ein Schlaganfall erkannt werden kann, desto weniger Nervenzellen sterben ab. Deshalb ist eine schnelle Diagnose, welche die Bestimmung des Zentrums des Schlaganfalls beinhaltet, und eine schnelle Durchführung einer Therapie wichtig. Die Diagnose kann durch ein Perfusions-CT¹ stattfinden. Die aufgenommenen Perfusions-CT Daten können mit der in

¹Dies ist eine funktionelle Untersuchung, bei der ein jodhaltiges Kontrastmittel injiziert wird und die Ausbreitung des Kontrastmittels mit mehreren kurzzeitig hintereinander gemachten CT Aufnahmen diagnostiziert wird [19].

der Arbeit entwickelten Visualisierung dargestellt werden. Dabei könnte das irreversibel geschädigtem Gewebe (Infarkt) und die Bereiche, die noch zu retten sind (Penumbra) als zweites Volumen in einer anderen Transferfunktion als das restliche Volumen dargestellt werden.

Bei der Traumatologie geht es darum, Patienten nach einem Unfall zu versorgen[14]. Dabei ist es wichtig schnell zu handeln und sich schnell einen Überblick zu verschaffen, welche Bereiche im Körper geschädigt sind und welche als erstes behandelt werden müssen. Liegen MRT oder CT Aufnahmen des Patienten vor, könnte die in der Arbeit entwickelte Visualisierung dafür genutzt werden, dass die Bereiche freigestellt werden, die geschädigt sind. Dadurch könnten sich die verschiedenen Ärzte einen Überblick über das Ausmaß der jeweiligen Schädigung machen.

Bei Tumoren in der Leber ist neben der RFA, die in Kapitel 2.1.2(Leberkrebs) bereits beschrieben wurde, auch eine Teilresektion der Leber möglich. Diese Behandlung kann mit der entwickelten Visualisierung dargestellt werden. Dabei kann der Tumor und die Blutgefäße der Leber mit der zweiten Transferfunktion eingefärbt werden. Die Leber könnte ausgeschnitten werden und mit der ersten Transferfunktion dargestellt werden. So kann geplant werden, wo die Gefäße in der Leber liegen und welcher Bereich der Leber rezidiert werden muss, damit der Tumor vollständig entfernt werden kann.

Die Behandlung von Hirn-Aneurysmen kann mit der Arbeit entwickelten Visualisierung dargestellt werden. Behandelt werden kann ein Aneurysma durch das *Clipping* oder *Coiling*. Beim *Coiling* wird an der Leistengegend ein Katheter eingeführt und dieser bis zum Gehirn geschoben [29]. Der Zugang und der Weg des Katheters bis ins Gehirn kann mit der entwickelten Visualisierung dargestellt werden.

6.2.2 Mögliche Erweiterungen

Die in Kapitel 3 vorgestellten Konzepte können noch weiter verbessert werden. Eine mögliche Verbesserung für die automatische Platzierung von Labels und eine mögliche Verbesserung für die Darstellung der synthetischen Objekte wird im folgenden beschrieben.

Labeling

Die implementierte automatische Platzierung der Labels in 3D kann noch erweitert werden, wodurch die Positionierung der Labels optimiert werden kann. Im aktuellen Stand werden die Positionen der Labels bei Hinzufügen oder Löschen eines Objektes nicht geändert. Außerdem kann es passieren, dass zwei Labels an genau der gleichen Position platziert werden. Dies passiert allerdings nur, wenn die zwei Labels genau die gleiche Position ihres Ankerpunkts besitzen. Der entwickelte Algorithmus zur automatischen Platzierung der Labels könnte wie in ein Algorithmus aus [4] erweitert werden, um die erwähnten Probleme zu beheben. In [4] wird ein Algorithmus für Randbeschriftungen vorgestellt. Dafür wurden verschieden Layouts zur die Randbeschriftungen entwickelt, wie z.B. Ring-Layout, Silhouetten-basiertes-Layout oder Radiales-Layout.

Darstellen von synthetischen Objekten

Für die synthetischen Objekte wurde in der vorliegenden Arbeit eine Beleuchtung entwickelt, die die Beleuchtungsmap des CRs nutzt. Für eine realere Darstellung dieser fehlen noch Schatten. Schatten sorgen dafür, dass räumlichen Relationen besser erkannt werden können. Der CR stellt Schatten im gerenderten Volumen dar, weshalb die tomographisch aufgenommenen Bilder als sehr realistisch angesehen werden. Zu einer realeren Darstellung der synthetischen Objekte kann das Darstellen von Schatten auf diesen Objekten oder Schatten, den diese Objekte auf andere Objekte der Szene werfen, beitragen. Die Realisierung eines *Shadow Mappings* ist in der vorgegebene Bearbeitungszeit der Arbeit nicht möglich gewesen, allerdings ist eine Idee vorhanden und wird im Folgenden noch kurz erwähnt.

Beim *Shadow Mapping* wird pro Lichtquelle der Szene eine 2D Abbildung, die *Shadow Map*, der Szene erstellt. Dafür wird von der Position der Lichtquelle die Szene betrachtet und alle Objekte abgebildet, die von der Lichtquelle aus gesehen werden. In Abbildung 6.3 ist dies schematisch dargestellt. In einem zweiten Schritt wird dann die generierte *Shadow Map* im Shader verwendet. Dieser Shader prüft für jeden Pixel, ob dieser im Schatten liegt oder nicht und rendert die Szene mit Schatten [3]. Einen anderen Ansatz für eine verbesserte Visualisierung der Oberfläche ist in [53] gegeben. In dieser Arbeit wird eine *Shading Map* erstellt, welche unterschiedliche Informationen über die Objekte einer Szene beinhaltet und diese durch unterschiedlich Gewichtung verschieden darstellen kann.

Für die Erstellung einer *Shadow Map* mit dem CR gerenderten Volumen und synthetische Objekten muss für das CR gerenderte Volumen ein Mesh generiert werden, da es sonst nicht als Objekt in der Szene auftaucht und nicht auf der *Shadow Map* abgebildet wird. Wie schon in Abschnitt 3.3.1(Abtasten der drei hellsten Bereiche) erwähnt wurde gibt es für den CR einen Modus, bei dem für die Beleuchtung nur drei Lichtquellen verwendet werden. Diese können an den Positionen, die in Abschnitt 3.3.1(Abtasten der drei hellsten Bereiche) abgetastet wurden, platziert werden und von diesen drei Lichtquellen kann dann jeweils eine *Shadow Map* erstellt werden. Die drei *Shadow Maps* können dann im Shader, welcher in Abschnitt 4.2.2(Umsetzung der Beleuchtung) beschrieben wurde, zum Rendern der Szene mit Schatten verwendet werden. Dadurch können Schatten gezeichnet werden, die das Volumen auf die synthetischen Objekte wirft. Damit Schatten, die die Objekte auf das Volumen werfen, entstehen, müsste der Shader im CR verändert werden.

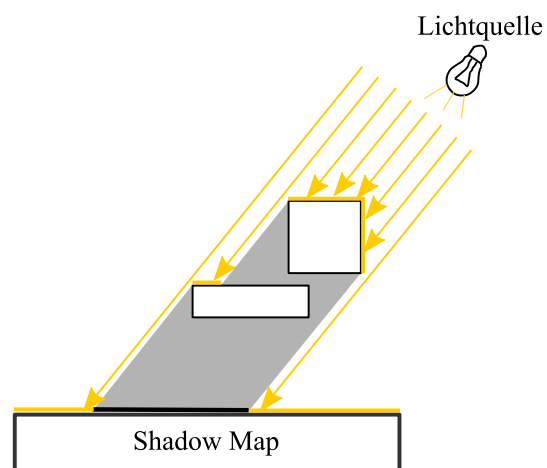


Abbildung 6.3: Prinzip Shadow Mapping (nach [3])

Anhang A

Fragebögen

Fragen zur Clipping Plane:

Wie werden die jeweiligen 3D Informationen des Objekts visualisiert?

Kategorie	Wichtigkeitsskala				
	sehr schlecht	schlecht	neutral	gut	sehr gut
Tiefeninformation					
Lage					
Größe					

Wie schnell können die dargestellten Informationen erfasst werden?

Wichtigkeitsskala				
sehr langsam	langsam	neutral	schnell	sehr schnell

Wie nützlich sind die Visualisierungen für die Patientenaufklärung vor Therapien?

Wichtigkeitsskala				
sehr nutzlos	nutzlos	neutral	nützlich	sehr nützlich

Fragen zur neuen Visualisierungsmethode

Wie werden die jeweiligen 3D Informationen des Objekts visualisiert?

Kategorie	Wichtigkeitsskala				
	sehr schlecht	schlecht	neutral	gut	sehr gut
Tiefeninformation					
Lage					
Größe					

Wie schnell können die dargestellten Informationen erfasst werden?

Wichtigkeitsskala				
sehr langsam	langsam	neutral	schnell	sehr schnell

Wie nützlich sind die Visualisierungen für die Patientenaufklärung vor Therapien?

Wichtigkeitsskala				
sehr nutzlos	nutzlos	neutral	nützlich	sehr nützlich

Allgemeine Anmerkungen:

Abbildung A.1: Fragebogen zur Smart Visibility kombiniert mit CR

Fragen zur Interaktion in 2D:

Wie intuitiv sind die einzelnen Interaktionen?

Kategorie	Wichtigkeitsskala				
	gar nicht intuitiv	nicht intuitiv	neutral	intuitiv	sehr intuitiv
Skalierung					
Rotation					
Translation					

Ist die Interaktion zielführend?

Wichtigkeitsskala				
gar nicht zielführend	nicht zielführend	neutral	zielführend	sehr zielführend

Fragen zur Interaktion in 3D:

Kategorie	Wichtigkeitsskala				
	gar nicht intuitiv	nicht intuitiv	neutral	intuitiv	sehr intuitiv
Skalierung					
Rotation					
Translation					

Ist die Interaktion zielführend?

Wichtigkeitsskala				
gar nicht zielführend	nicht zielführend	neutral	zielführend	sehr zielführend

Allgemeine Anmerkungen:

Abbildung A.2: Fragebogen zur Interaktion

Fragen zur Beleuchtung:

Wie homogen passt die Darstellung des Objekts zum Volumen?

	Wichtigkeitsskala				
	sehr schlecht	schlecht	neutral	gut	sehr gut
1.Bild					
2.Bild					
3.Bild					
4.Bild					
5.Bild					
6.Bild					

Mit welchem Bild kann man die Therapieplanung einem Patienten am besten zeigen?

1.Bild 2.Bild 3.Bild

Mit welchem Bild kann man die Therapieplanung einem Patienten am besten zeigen?

4.Bild 5.Bild 6.Bild

Allgemeine Anmerkungen:

Abbildung A.3: Fragebogen zur Beleuchtung

Literaturverzeichnis

- [1] *Barycentre*. <https://www.encyclopediaofmath.org/index.php/Barycentre>
- [2] *Eigen value*. http://www.encyclopediaofmath.org/index.php?title=Eigen_value&oldid=30984
- [3] *Tutorial 16 : Shadow mapping*. <http://www.opengl-tutorial.org/intermediate-tutorials/tutorial-16-shadow-mapping/>
- [4] ALI, K. ; HARTMANN, K. ; STROTHOTTE, T.: Label Layout for Interactive 3D Illustrations. In: *Journal of the WSCG* Bd. 13:1-8, 2005
- [5] BARNES, B. ; BERTZ, J. ; BUTTMANN-SCHWEIGER, N. ; FIEBIG, J. ; JORDAN, S. ; KRAYWINKEL, K. ; NIEMANN, H. ; NOWOSSADECK, E. ; POETHKO-MÜLLER, C. ; PRÜTZ, F. ; RATTAY, P. ; SCHÖNFELD, I. ; STARKER, A. ; WIENECKE, A. ; WOLF, U.: *Bericht zum Krebsgeschehen in Deutschland 2016*. Berlin: Robert Koch-Institut, 2016
- [6] BAUER, K. ; ENNKER, J.: *Herzklappenchirurgie: Ein Patientenratgeber*. Steinkopff, 2005 (Operationen am Herzen)
- [7] BRUCKNER, S. ; GRIMM, S. ; KANITSAR, A. ; GRÖLLER, E.: Illustrative Context-Preserving Volume Rendering. In: *Proceedings of EuroVis 2005*, 2005, S. 69–76
- [8] BRUCKNER, S. ; GRÖLLER, E.: VolumeShop: An Interactive System for Direct Volume Illustration. In: *Proceedings of IEEE Visualization 2005*, 2005, S. 671–678
- [9] CHRISTEN, M.: *Per Fragment Lighting*. <https://www.opengl.org/sdk/docs/tutorials/ClockworkCoders/lighting.php>
- [10] CONNER, B.D. ; SNIBBE, S.S. ; HERNDON, K.P. ; ROBBINS, D.C. ; ZELEZNIK, R.C. ; DAM, A. van: Three-dimensional Widgets. In: *Proceedings of the 1992 Symposium on Interactive 3D Graphics*, 1992, S. 183–188
- [11] DANIEL, W.G. ; BAUMGARTNER, H. ; GOHLKE-BÄRWOLF, C. ; HANRATH, P. ; HORSTKOTTE, D. ; KOCH, K.C. ; MÜGGE, A. ; F.A. FLACHSKAMPF, H.J. S.: *Pocket-Leitlinien: Klappenvitien im Erwachsenenalter*. Vorstand der Deutschen Gesellschaft für Kardiologie – Herz- und Kreislaufforschung e.V., 2007
- [12] DEUSSEN, O.: *Lokale Beleuchtungsmodelle*. https://www.inf.tu-dresden.de/content/institutes/smt/cg/teaching/lectures/CG3WS0203/secure/lokaleBeleuchtung_script.pdf

- [13] DONHAUSER, J.: *Hintergrundinformation: Wie Cinematic Volume Rendering Technique den Blick auf klinische Bilder verändert.* <https://www.siemens.com/press/pool/de/events/2016/healthcare/2016-11-rsna/hintergrund-cinematic-rendering-d.pdf>
- [14] DUDENREDAKTION: „Traumatologie“ auf Duden online. <http://www.duden.de/node/687076/visions/1614119/view>
- [15] FRAUNHOFER MEVIS: *Image Processing Research and Development.* <https://www.mevislab.de/mevislab/>
- [16] FRITSCHER-RAVENS, A. ; FEUSSNER, H. ; KÄHLER, G. ; MATHES, K. ; MEINING, A. ; HOCHBERGER, J. ; MEIER, P. ; DELIUS, S. von ; RENTELN, D. von ; WILHELM, D. ; BURGHARDT, J. ; BERNHARDT, J. ; LAMADE, W. ; MAGDEBURG, R. ; MEIER, P. ; FUCHS, M. Vassiliou and K. u. a.: State of NOTES development in Germany: Status report of the D-NOTES-Congress 2011th, 2012, S. 325–330
- [17] HERZBERG, D.: *On-X Heart Valve.* <https://grabcad.com/library/on-x-heart-valve>
- [18] HODGES, E.R.S.: *The Guild Handbook of Scientific Illustration.* Wiley, 1988
- [19] INSTITUT UND POLIKLINIK FÜR NEURORADIOLOGIE: *CT-Perfusion.* <https://www.uniklinikum-dresden.de/de/das-klinikum/kliniken-polikliniken-institute/nra/patienteninfo/diagnostische-verfahren/ct-perfusion>
- [20] KAJIYA, J.T.: *The Rendering Equation*, 1986, S. 143–150
- [21] KAUCZOR, H.-U.: *Radiofrequenzablation (RFA).* https://www.klinikum.uni-heidelberg.de/fileadmin/radiologie/radiodiagnostik/Funktionsbereiche/Interventionsgruppe/RFA_Pat.Info.1.pdf
- [22] KLAWONN, F.: *Grundkurs Computergrafik mit Java: Die Grundlagen verstehen und einfach umsetzen mit Java 3D.* Vieweg+Teubner Verlag, 2010 (Vieweg Studium)
- [23] KÜNZEL, U.: *Hintergrundinformation: Cinematic Rendering für die Chirurgie soll die Planung operativer Eingriffe effizienter und präziser machen.* <https://www.siemens.com/press/pool/de/feature/2017/healthineers/2017-06-cinematic-rendering/hintergrund-cinematic-rendering-chirurgie-d.pdf>
- [24] KRÜGER-FRANKE, M.: *Engelhardt Lexikon Orthopädie und Unfallchirurgie.* <http://www.lexikon-orthopaedie.com/pdx.pl?dv=0&id=01734>
- [25] KROES, T. ; POST, F. ; BOTHA, C. P.: Exposure Render: An Interactive Photo-Realistic Volume Rendering Framework. In: *PLOS ONE* Bd. 7, Public Library of Science, 2012, S. 1–10
- [26] KUBISCH, C. ; TIETJEN, C. ; PREIM, B.: GPU-based smart visibility techniques for tumor surgery planning, 2010, S. 667–678

- [27] LAMBERT, J.H. ; ANDING, E.: *Lamberts Photometrie: Photometria, sive De mensura et gradibus luminus, colorum et umbrae (1760)*. W. Engelmann, 1892 (Lamberts Photometrie: Photometria, sive De mensura et gradibus luminus, colorum et umbrae)
- [28] LIKERT, R.: A technique for the measurement of attitudes., 1932, S. 1–55
- [29] LINN, J.: *Aneurysma-Coiling*. http://www.klinikum.uni-muenchen.de/Abteilung-fuer-Neuroradiologie/de/Was_ist_Neuroradiologie_/nrad_therapie/nrad_ther_coil/index.html
- [30] LORENSEN, W.E. ; CLINE, H.E.: Marching Cubes: A High Resolution 3D Surface Construction Algorithm. In: *SIGGRAPH Comput. Graph.* Bd. 21, ACM, 1987, S. 163–169
- [31] LORENSEN, W.E. ; CLINE, H.E.: Marching Cubes: A High Resolution 3D Surface Construction Algorithm. In: *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques*, ACM, 1987 (SIGGRAPH '87), S. 163–169
- [32] MANTIUK, R.K. ; MYZKOWSKI, K. ; SEIDEL, H.-P.: High Dynamic Range Imaging. In: *Wiley Encyclopedia of Electrical and Electronics Engineering*, 2016
- [33] MARTIN, O.: Licht und Schatten / Institut für Visualisierung und Interaktive Systeme Universität Stuttgart. Version: 2010. <http://www.vis.uni-stuttgart.de/plain/seminare/computerspiele/olegmartin/ausarbeitung.pdf>. 2010. – Forschungsbericht
- [34] MARTSCHINKE, J.: *Physikbasierte Volumenvisualisierung mit Irradiance Caching*, Friedrich-Alexander-Universität Erlangen-Nürnberg, Masterarbeit, 2016
- [35] MEVIS MEDICAL SOLUTIONS: *Getting Started*. <http://mevislabdownloads.mevis.de/docs/current/MeVisLab/Resources/Documentation/Publish/SDK/GettingStarted/>
- [36] MOGALLE, K.: *Automatische Platzierung von klinischen Annotationen in CT-Bildern*, Otto-von-Guericke-Universität Magdeburg, Bachelorarbeit, 2012
- [37] MOGALLE, K. ; TIETJEN, C. ; SOZA, G. ; PREIM, B.: Constrained Labeling of 2D Slice Data for Reading Images in Radiology. In: *VCBM*, 2012
- [38] MÜHLER, K. ; P., Bernhard: Automatic Textual Annotation for Surgical Planning. In: *Proceedings of the Vision, Modeling, and Visualization Workshop 2009, November 16-18, 2009, Braunschweig, Germany*, 2009, S. 277–284
- [39] MÜLLER-STICH, B.P. ; BÜCHLER, M.W.: Operationssaal der Zukunft. In: *Der Chirurg* Bd. 87, 2016, S. 999–1001
- [40] NISCHWITZ, A. ; FISCHER, M. ; HABERÄCKER, P. ; SOCHER, G.: *Computergrafik und Bildverarbeitung: Band I: Computergrafik*. Vieweg+Teubner Verlag, 2012

- [41] OLANO, M. ; LASTRA, A.: A Shading Language on Graphics Hardware: The Pixel-flow Shading System. In: *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques*, 1998 (SIGGRAPH '98), S. 159–168
- [42] ONMEDA-REDAKTION: *Schlaganfall (Apoplex, Hirnschlag, Hirninfarkt, Insult)*. <http://www.onmeda.de/krankheiten/schlaganfall.html>
- [43] PABST, C.: *Magnetresonanz-Tomographie: Lernskript für Mediziner Grundlagen der Magnetresonanz-Tomographie*. http://www.ukgm.de/ugm_2/deu/umr_rdi/Teaser/Grundlagen_der_Magnetresonanztomographie_MRT_2013.pdf
- [44] PREIM, B.: *Visualization in Medicine: Theory, Algorithms, and Applications (The Morgan Kaufmann Series in Computer Graphics)*. Morgan Kaufmann, 2007
- [45] PREIM, B. ; RITTER, A. ; STROTHOTTE, T. ; POHLE, T. ; BARTRAM, L. ; FORSEY, D.: *Consistency of Rendered Images and Their Textual Labels*. 1995
- [46] ROGERS, D.F. ; EARNSHAW, R.: *Computer Graphics Techniques: Theory and Practice*. Springer-Verlag New York, 1990
- [47] RUSH, P.: *Valve disease*. <https://www.allinahealth.org/health-conditions-and-treatments/health-library/patient-education/helping-your-heart/types-of-heart-problems/what-can-go-wrong-with-your-heart/valve-disease/>
- [48] SCHERÜBL, H.: *Therapie von Leberkrebs*. <https://www.krebsgesellschaft.de/onko-internetportal/basis-informationen-krebs/krebsarten/andere-krebsarten/leberkrebs/therapie.html#RFA>
- [49] SHAHLAEI, D. ; BLANZ, V.: Realistic inverse lighting from a single 2D image of a face, taken under unknown and complex lighting. In: *11th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition*, 2015, S. 1–8
- [50] STEPHAN, O.: *Beleuchtungsmodelle der Computer Grafik*. <http://olli.informatik.uni-oldenburg.de/i-light/tutorial.html>
- [51] TAPPENBECK, A. ; PREIM, B. ; DICKEN, V.: Distance-based transfer function design: Specification Methods and Applications. In: *SimVis*, SCS Publishing House e.V., 2006, S. 259–274
- [52] TIETJEN, C.: *Illustrative Visualisierungstechniken zur Unterstützung der präoperativen Planung von chirurgischen Eingriffen*, Otto-von-Guericke-Universität Magdeburg, Diss., 2009
- [53] TIETJEN, C. ; PFISTERER, R. ; BAER, A. ; GASTEIGER, R. ; PREIM, B.: Hardware-Accelerated Illustrative Medical Surface Visualization with Extended Shading Maps. In: *Smart Graphics: 9th International Symposium, SG 2008, Rennes, France, August 27-29, 2008. Proceedings*. Berlin, Heidelberg : Springer Berlin Heidelberg, 2008, S. 166–177

- [54] TÖNNIES, K.D.: *Grundlagen der Bildverarbeitung*. München : Pearson Studium, 2005
- [55] TÖNNIS, K.D.: *Guide to Medical Image Analysis*. Springer, 2012
- [56] VIOLA, I. ; GRÖLLER, E.: Smart Visibility in Visualization. In: *IEEE Visualization*. IEEE Computer Society Press, 2005, S. 139–145
- [57] VIOLA, I. ; KANITSAR, A. ; GRÖLLER, E.: Importance-Driven Volum Rendering. In: *IEEE Visualization*. IEEE Computer Society Press, 2004
- [58] VRIES, J. de: *Diffuse irradiance*. <https://learnopengl.com/#!PBR/IBL/Diffuse-irradiance>

