



BEUTH HOCHSCHULE FÜR TECHNIK BERLIN  
University of Applied Sciences

BEUTH HOCHSCHULE FÜR TECHNIK

ABSCHLUSSARBEIT MASTER MEDIENINFORMATIK

# **mARt: Interaktive Darstellung von MRT Daten in AR**

*Viola Jertschat*

betreut von

Prof. Dr.-Ing. Kristian HILDEBRAND

12. Februar 2019

## **Zusammenfassung**

...

## Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>4</b>
1.1	Motivation . . . . .	4
1.2	Zielsetzung . . . . .	5
1.3	Struktur dieser Arbeit . . . . .	5
<b>2</b>	<b>Verwandte Arbeiten und Technologien</b>	<b>8</b>
2.1	AR in der Medizin . . . . .	8
2.2	Visualisierung von MRT-Daten in AR/VR . . . . .	8
2.3	MRT Volume Rendering . . . . .	8
2.4	Interaktion in AR . . . . .	8
<b>3</b>	<b>Aktueller Stand der Technik</b>	<b>9</b>
3.1	MRT . . . . .	9
3.1.1	Verfahren . . . . .	9
3.1.2	Datenverarbeitung . . . . .	9
3.1.3	Abgrenzung CT . . . . .	10
3.1.4	Datenformate . . . . .	10
3.2	Volumendaten . . . . .	10
3.3	Volume Rendering . . . . .	10
3.3.1	Volumetrisches Ray-Casting . . . . .	11
3.3.2	Texture Based Volume Rendering . . . . .	12
3.3.3	Shear-Warp . . . . .	12
3.4	Oberflächen Generierung (Weitere Methoden zur dreidimensionalen Darstellung von MRT-Daten) . . . . .	13
3.4.1	Marching Cubes . . . . .	13
3.4.2	Voxel . . . . .	15
3.4.3	Andere . . . . .	15
3.5	Vergleich der Methoden im Überblick . . . . .	15
3.6	AR und VR (MR?) . . . . .	15
3.6.1	Augmented Reality . . . . .	15
3.6.2	Virtual Reality . . . . .	16
3.7	Verarbeitung von MRT-Daten . . . . .	18
3.7.1	Arbeit eines Neurologen . . . . .	18
3.7.2	Software in der Radiologie . . . . .	18
3.8	Interaktion in AR/VR . . . . .	19
3.8.1	Eingabe Modularitäten . . . . .	19
3.8.2	Integrierte Nutzereingaben . . . . .	19
3.8.3	Leap Motion . . . . .	19
3.8.4	Andere . . . . .	19
3.8.5	Interaktionsdesign . . . . .	19
<b>4</b>	<b>Anforderungsanalyse</b>	<b>20</b>

4.1	Interviews . . . . .	20
4.1.1	Anforderungen aus Interviews . . . . .	20
4.2	User Stories . . . . .	21
4.3	Technische Anforderungen . . . . .	23
<b>5</b>	<b>Konzept</b>	<b>24</b>
5.1	3D Darstellung . . . . .	24
5.2	Darstellung des gekennzeichneten Bereichs . . . . .	24
5.3	Endgerät . . . . .	24
5.4	Interaktion/ UI . . . . .	25
5.4.1	Interaktion AR . . . . .	25
5.4.2	Interaktion VR . . . . .	25
5.5	Unterstützung von Dateiformaten . . . . .	25
<b>6</b>	<b>Implementierung</b>	<b>26</b>
6.1	Aufbau Struktur des Projektes . . . . .	26
6.1.1	Kombination von Leap Motion mit Vive/Hololens . . . . .	26
6.1.2	Unity Projekt . . . . .	26
6.2	Interaktion Hololens . . . . .	26
6.2.1	HoloToolkit . . . . .	26
6.3	Interaktion Vive . . . . .	26
6.3.1	Leap Motion . . . . .	26
6.4	Volume Rendering . . . . .	26
6.4.1	Shader in Unity . . . . .	26
6.4.2	Ray Casting . . . . .	26
6.4.3	Transferfunktion . . . . .	28
6.4.4	Illumination . . . . .	28
6.4.5	Gradientenberechnung . . . . .	29
<b>7</b>	<b>Evaluation</b>	<b>30</b>
7.1	Vergleich mit Anforderungen . . . . .	30
7.2	Nutzertest . . . . .	30
7.3	Ergebnisse . . . . .	30
<b>8</b>	<b>Zusammenfassung und Fazit</b>	<b>31</b>
8.1	Zusammenfassung . . . . .	31
8.2	Ausblick . . . . .	31
8.3	Fazit . . . . .	31

## Abbildungsverzeichnis

1	Die 15 verschiedenen Möglichkeiten, wie eine Polygonfläche einen Würfel in zwei Bereiche teilen kann. . . . .	14
---	---	----

# Kapitel 1

## Einleitung

Der Einsatz von MRTs ermöglicht es Ärzten einen Einblick in das innere des menschlichen Körpers zu erlangen, ohne diesen dabei zu verletzen. Sie erhalten Bilder innerer Organe, anhand derer sich dessen Aufbau und Funktionalität beobachten lassen. Aber auch mögliche Fehlfunktionen oder Anomalien können so erfasst werden. So können auf MRTs-Scans gebrochene Knochen, innere Verletzungen oder Schlaganfälle erkannt und beurteilt werden. Im Fall von Schlaganfällen ist hierbei nicht nur eine Diagnose möglich sondern sogar eine Prävention. In beiden Fällen ist es unabdingbar, dass der Arzt eine möglichst umfassende Vorstellung von der Struktur des Gehirns des Patienten und vor allem von den vom Schlaganfall betroffenen Bereichen hat. Nur wenn dies der Fall ist, kann eine sinnvolle Therapie angewandt werden. Diese Arbeit stellt die Möglichkeit vor den Umgang mit MRT-Daten anschaulicher und intuitiver zu gestalten, um somit die Arbeit von Radiologen im Bereich der Schlaganfallprävention zu erleichtern und die Gesundheit ihrer Patienten zu verbessern.

### 1.1 Motivation

Um MRT-Scans zu studieren benutzen Ärzte in der Regel speziell dafür entwickelte Software. Diese stellt das Gehirn meist aus der Sicht von drei verschiedenen Achsen dar, sodass es von allen Seiten zusehen ist. Auf diesen Achsen kann der Nutzer die Ansicht dann durch die Schichten des Scans bewegen. Die aktuelle Position der gerade angezeigten Schicht wird in jeder der anderen Achsenansichten farbig eingezeichnet, um den Nutzer ein möglichst umfassendes Bild des gescannten Gehirns zu vermitteln. Ein Beispiel für die Oberfläche solch einer eben beschriebenen Software ist in Abbildung ?? zu sehen.

Anwendungen dieser Art sind weit verbreitet. Die zweidimensionale Ansicht, in der die Bilder vorliegen können allerdings eine falsche Vorstellung von der vorliegenden Situation schaffen. Durch die Reduzierung um eine Dimension entsteht ein verzerrtes Bild des Gehirns. Die Darstellung der verschiedenen Achsen auf den Scans soll den Arzt bei der Orientierung unterstützen. Trotzdem muss dieser immer seinen Fokus zwischen zwei oder mehr Bildern wechseln und über ein gewisses räumliches Vorstellungsvermögen besitzen, um das Gesamtbild des Gehirns in seinem Geist zu rekonstruieren. Dies ist ein kognitiver Aufwand, der Ärzte zusätzlich belastet, während sie sich darauf konzentrieren Anomalien in den Scans eines Patienten zu erkennen und einzuschätzen. Wird auf einem Scan ein Schlaganfall entdeckt, ist es durch die Abstraktion des Organs schwierig eine korrekte Vorstellung von der Größe und Lage des betroffenen Bereichs zu bekommen, da jeweils nur eine Schicht des Gehirns sichtbar ist. Eine dreidimensionale Ansicht des gescannten Gehirns würde einen sehr viel deutlicheren Einblick in den Zustand des Patienten liefern. Vor allem der vom Arzt gekennzeichnete betroffene Bereich wäre in 3D um einiges anschaulicher. Dies ist nicht nur für den behandelnden Arzt hilfreich. Durch die klare und eindeutige Darstellung fällt es auch leichter den anderen die Situation zu erläutern. Dies trifft auf Patienten zu oder auch auf andere Ärzte, die der behandelnde Arzt eventuell in den Fall

mit einbeziehen möchte. Schließlich würde eine 3D-Darstellung auch das Verständnis in Lernzwecken begünstigen. Da der betroffene Bereich in einer 3D-Darstellung auf einen Blick erfasst werden kann, eignet sie sich außerdem, um den direkten Vergleich zwischen zwei Zuständen zu ziehen. So fiel es leichter beispielsweise die Größe des Bereichs vor und nach einer Therapie gegenüberzustellen, um deren Erfolg zu demonstrieren oder zu beurteilen.

Die eben beschriebenen Vorteile einer 3D-Darstellung würden theoretisch auch in einer Bildschirmanwendung gelten. Allerdings würde das Potenzial der Räumlichkeit damit nicht vollkommen ausgeschöpft werden. Weiterhin ist es das Ziel die Interaktion mit den Daten möglichst intuitiv zu gestalten, damit der Fokus des Nutzers auf den Bildern selbst liegen kann. Durch die Platzierung im Raum wäre die Darstellung anschaulicher und der Nutzer könnte direkt mit ihr interagieren. Um dies zu erreichen bieten sich die virtuelle oder augmentierte Realität an. Eine AR-Anwendung scheint für diesen Fall allerdings besser geeignet zu sein. Durch die Umsetzung der Darstellung in AR wird die Interaktion mit den Daten direkter und intuitiver, da der Nutzer seine Hände als Eingabemedium benutzen kann. Dies macht die Verwendung der Software leichter zugänglich, was unter anderem für Lernzwecke dienlich sein kann. Zudem wird die Nutzung der Anwendung dadurch interessanter und unterhaltsamer. Gleichzeitig ist durch die kabellosen, tragbaren AR-Headsets eine höhere Mobilität gegeben, als durch einen Rechner. Dadurch kann die Anwendung unabhängig von der Umgebung überall zum Einsatz kommen.

AR-Anwendungen entwickeln sich stetig weiter und werden in der Zukunft einen immer größeren Teil des Alltags einnehmen. Diese Entwicklung wirkt sich auch auf den medizinischen Bereich aus. Ärzte sind sich der neuen Möglichkeiten bewusst und sind daran interessiert, in welchen Einsatzgebieten man einen Nutzen aus diesen ziehen kann. (Studie) Eine Anwendung wie mARt eignet sich gut, um den praktischen Einsatz von AR prototypisch zu testen.

## **1.2 Zielsetzung**

Ziel dieser Arbeit ist es, die Möglichkeiten der Darstellung von und Interaktion mit MRT-Daten in AR untersuchen. Der Fokus liegt dabei auf der Darstellung von Gehirnscans, die in der Schlaganfallprävention und -diagnose verwendet werden. Hierzu soll eine prototypische Anwendung konzipiert und implementiert werden, die die eben genannten Möglichkeiten demonstriert: mARt. Die MRT-Bilder werden innerhalb einer AR-Anwendung dreidimensional dargestellt. Außerdem soll eine möglichst intuitive Interaktion mit der Darstellung ermöglicht werden. Um eine nützliche Anwendung zu entwickeln, die den Anforderungen eines Einsatzes im Arbeitsfeld eines Radiologen entspricht, werden Interviews mit einem Radiologen geführt werden. Aus diesen wird dann die nötige Funktionalität der Anwendung abgeleitet. Die Anwendung ist nicht als einsetzbares Produkt zu verstehen sondern eher als Prototyp, der die Nützlichkeit und das Potenzial des Programms beweisen soll. Der Nutzen der Anwendung wird am Ende der Arbeit evaluiert.

## **1.3 Struktur dieser Arbeit**

Zuerst wird in Kapitel 2 betrachtet, welche anderen Arbeiten bereits existieren, die sich mit einem ähnlichem Thema befassen oder die inhaltlich die Thematik dieser Arbeit berühren. Im Kapitel ?? werden dann theoretische Grundlagen zu Methoden und Techniken erläutert, die für das Verständnis dieser Arbeit notwendig bzw. hilfreich sind. Um die genaue Funktionalität und Umfang der zu implementierenden Anwendung festzustellen, werden in Kapitel 4 die Interviews mit den bereits erwähnten Radiologen ausgewertet und daraus User Stories und schließlich eine Anforderungsliste erstellt. Anhand dieser Anforderungen wird in Kapitel 5 ein theoretischer Entwurf der Anwendung ausgearbeitet, indem Methoden und Techniken, sowie die Benutzung des Programms diskutiert und festgelegt werden. Die Umsetzung des entwickelten Konzeptes wird schließlich in Kapitel 6 beschrieben. Dabei wird auch Hürden eingegangen, die im Rahmen

dieser auftraten. Die Anwendung wird anschließend getestet und mit den zuvor gestellten Anforderungen verglichen. Die Ergebnisse werden in Kapitel 7 beschrieben. In Kapitel ?? werden die Schwerpunkte der Arbeit noch einmal zusammengefasst und mögliche Weiterentwicklungen in der Zukunft werden diskutiert.



-; Zusammenführen mit Grundlagen

## Kapitel 2

### Verwandte Arbeiten und Technologien

#### 2.1 AR in der Medizin

#### 2.2 Visualisierung von MRT-Daten in AR/VR

[**SORENSEN2001193**] - Erstellen eines Meshes anhand der MRT Daten - Selbst entwickelte VR Umgebung, die Mesh darstellt und Interaktion erlaubt - beschränkt sich auf Herz - 2001

[**PMID:12635021**] - selbstgebautes AR-Headset - markergestützt - stellt MRT-Bilder auf Kopfattrappe dar - soll Interventionen unterstützen - 2003

[**Watts:2017:PAR:3139131.3141198**] - Projektion von 3D-MRT Bildern auf Patient - Volume Rendering - Aufbau der AR-Hardware steht im Vordergrund

[**Ghoshal:2017:STV:3173519.3173527**] -

#### 2.3 MRT Volume Rendering

#### 2.4 Interaktion in AR

## Kapitel 3

### Aktueller Stand der Technik

#### 3.1 MRT

Die Abkürzung MRT steht für Magnetresonanztomographie. Das bildgebende Verfahren, das auch Kernspintomografie genannt wird, wird in der Radiologie verwendet, um Abbildungen innerer Organe zu erzeugen. Während der Durchführung einer MRT wird der Patient in ein MRT-System geschoben, das einer großen Röhre gleicht. Er sollte sich für die Dauer des MRTs möglichst wenig bewegen, um klare Bilder zu erhalten.

##### 3.1.1 Verfahren

Um die inneren Organe eines Patienten zu visualisieren, werden kleinste Teilchen seines Körpers in Bewegung versetzt, die gemessen werden kann. Im Falle einer MRT handelt es sich dabei um Wasserstoffprotonen. Diese haben eine Eigendrehung um sich selbst, den sogenannten Kernspin. Durch ihre positive Ladung, die durch den Kernspin in Bewegung ist, besitzen die Protonen weiterhin ein eigenes Magnetfeld, welches messbar ist. Während einer MRT wird mit einer Spule, die in dem MRT-System verbaut ist um den Körper des Patienten ein Magnetfeld erzeugt. Die Kernspin-Achsen der Wasserstoffprotonen richten sich an diesem aus. Anschließend wird in das Magnetfeld ein Hochfrequenzimpuls, die Larmorfrequenz eingestrahlt. Durch diesen Impuls findet eine Synchronisation der Protonen statt, wobei einige um  $180^\circ$  gedreht werden. Kurz danach laufen die Protonen wieder auseinander und richten sich wieder am Magnetfeld aus. Dadurch, dass alle Protonen in dieselbe Richtung zeigen (phasengleich sind), verstärken sie gegenseitig das Signal, das sie abgeben. Das Signal wird schwächer, sobald sie wieder auseinander laufen (Dephasierung). Die Zeit, die die Protonen brauchen, um sich wieder am Magnetfeld auszurichten wird als Relaxionszeit bezeichnet. Dabei wird zwischen T1- und T2-Relaxtion unterschieden. Die Relaxionszeit ist dabei abhängig von der Zusammensetzung des umgebenden Gewebes. Das gemessene MR-Signal, das durch diese beeinflusst wird, ist also für verschiedene Gewebearten verschieden stark.

##### 3.1.2 Datenverarbeitung

Da die MRT ein dreidimensionales Objekt abbilden soll, werden mit dem eben beschriebenen Verfahren die Werte auch der XY-Ebene erfasst. Das Gehirn wird in Z-Richtung in einzelne Schichten unterteilt. Eine Schicht ist dabei gewissermaßen ein Bild mit X- und Y-Koordinaten, von dem viele hintereinander gehängt werden. Diese Unterteilung in Schichten wird erreicht, indem Z-Gradientenspulen verwendet werden, die das Magnetfeld inhomogen machen und entsprechend dem Z-Gradienten zu einer Seite hin abfallen lassen. So kann jeder Z-Schicht eine bestimmte Stärke im Magnetfeld zugewiesen werden. Es wird immer nur eine Schicht auf einmal gescannt und verarbeitet.

Die X- und Y-Werte einer Schicht repräsentieren allerdings nicht, wie bei einem Bild Koordinaten, die der Anordnung der jeweils betrachteten Punkte in der Welt entsprechen. Stattdessen

bildet der X-Wert die Frequenz und der Y-Wert die Phase ab. Wie für den Z-Wert werden auch hier Gradienten gebildet und auf das Magnetfeld gelegt. Der X-Gradient verläuft von links nach rechts und sorgt dafür, dass die Larmorfrequenz in dieser Richtung zunimmt, sodass die jeder Punkt seine eigene Frequenz hat. Der Y-Gradient, der senkrecht verläuft, beeinflusst auf dieselbe Weise die Phasen einer Schicht. Er wird dabei nur kurz nach dem Einstrahlen des Hochfrequenz-Impulses eingeschaltet, wenn sich die Protonen bereits ausgerichtet haben.

Für jeden Punkt gibt es also eine Magnetfeldstärke (Z), eine Frequenz (X) und eine Phase (Y). Diese Werte aller Punkte werden in einer Matrix gespeichert, die K-Raum genannt wird. Die Matrix entspricht allerdings noch nicht der bildlichen Darstellung, die angestrebt wird, da die Werte eine andere Bedeutung haben. Deshalb werden sie mit Hilfe der Fouriertransformation in lesbare Bilddaten umgewandelt, die die entsprechenden Organe schichtenweise abbilden.

### 3.1.3 Abgrenzung CT

Eine von der Durchführung ähnliche Methode zur Abbildung des Körperinneren, ist die Computertomographie. Die Verfahren unterscheiden sich jedoch. Denn bei einer CT wird der Patient schichtenweise geröntgt. D.h. sein Körper wird mit Röntgenstrahlung beschossen, die je nach Gewebe, auf das sie treffen unterschiedlich stark abgeschwächt werden, was dann gemessen wird. Die so entstandenen "Querschnitte" des Körpers werden anschließend mit Hilfe eines Computers zu einem dreidimensionalen Bild zusammengesetzt.

Die CT ist deutlich kürzer als eine MRT. Deshalb wird sie oft bei Notfällen verwendet. Allerdings wird der Patient dabei auch der Belastung von radioaktiver Strahlung ausgesetzt ist, die stärker ist als beim normalen Röntgen. Außerdem ist können Weichteile mit einer MRT besser dargestellt werden. Sie eignet sich also mehr zur Untersuchung des Gehirns.

### 3.1.4 Datenformate

MRT-Bilder werden meist in Dateiformaten gespeichert, die in der Medizin üblich sind. Dazu gehören nifti oder DICOM, welches z.B. neben den Bildern auch Patientendaten speichert. Allerdings 16 bit int images, pvm ? Standart?

## 3.2 Volumendaten

Das Volumen setzt sich aus mehreren dreidimensionalen Bildpunkten zusammen. Diese werden Voxel genannt, vergleichbar mit Pixeln in zweidimensionalen Bildern.

Voxel

3D Texturen

Isofläche

Klassifikation

## 3.3 Volume Rendering

Wie in Kapitel 1.1 erläutert, sollen die MRT-Bilder, die der Neurologe untersucht in mARt als dreidimensionales Volumen dargestellt werden. Volume Rendering bezeichnet die Darstellung eines dreidimensionalen Volumens, meist durch ein Skalarfeld repräsentiert, auf einer zweidimensionalen Bild. MRT-Bilder, die in Graustufen vorliegen bilden ein solches Skalarfeld. Dadurch, dass das Volumen aus Voxeln gebildet wird, gibt es keine Oberfläche, die das abzubildende Objekt beschreibt. Die Darstellung erfolgt deshalb anhand eines optischen Modells. Jedem Wert im Datensatz werden dazu optische Eigenschaften zugewiesen, im Allgemeinen Farbe und Opazität.

Indem Strahlen durch das Volumen geschossen werden, werden diese Eigenschaften mit einander verrechnet, was schließlich zur Abbildung des gesamten Volumen führt. Diese Technik wird im Abschnitt 3.3.1 genauer beschrieben. Die optischen Eigenschaften eines Voxels sind abhängig von seinem Isowert, sowie der verwendeten Transferfunktion und dem Shading.

Eine Transferfunktion besteht meistens aus einer Textur, aus der für jeden Isowert eine Farbe und Opazität gelesen werden kann. Auf diese Weise können Voxel, die einem bestimmten Bereich oder z.B. Gewebe angehören hervorgehoben oder ausgeblendet werden. Eine gute Transferfunktion zu implementieren ist sehr schwierig, da die korrekten Werte oft nur durch Ausprobieren gefunden werden und von dem jeweiligen Datensatz abhängen.

Das Shading bestimmt unter anderem die Beleuchtung eines Voxels und kann dessen Eigenschaften somit zusätzlich zur Transferfunktion verändern. Hierbei können verschiedene Beleuchtungsmodelle verwendet werden. Bei einer lokalen Beleuchtung der einzelnen Voxel kommt hierzu meist das Phong-Beleuchtungsmodell zum Einsatz.

Phong Normalen Gradienten etc... ? Segmentierung

Wie bereits erwähnt, werden die einzelnen Werte entlang eines Strahls mit einander verrechnet. Dabei werden sie in der Regel aufeinander addiert. Dieser Vorgang wird Komposition genannt. Dabei ist es von Relevanz, in welcher Reihenfolge die Werte durchlaufen werden. Wird das Volumen von vorne nach hinten durchlaufen, erfolgt die Komposition wie folgt:

$$\hat{C}_i = (1 - \hat{A}_{i-1})C_i + \hat{C}_{i-1}$$

$$\hat{A}_i = (1 - \hat{A}_{i-1})A_i + \hat{A}_{i-1}$$

Wobei  $\hat{C}_i$  die Farbe und  $\hat{A}_i$  die Transparenz der Farbe des vordersten Voxels ist. Sollen die Werte von hinten nach vorne addiert werden, wird dies durch die folgende Formel beschrieben.

$$\hat{C}_i = C_i + (1 - \hat{A}_i)\hat{C}_{i+1}$$

$$\hat{A}_i = A_i + (1 - \hat{A}_i)\hat{A}_{i+1}$$

Skalarfeld Optical Model Classification? Lightning

Es gibt verschiedene Techniken, von denen die wichtigsten im Folgenden erläutert werden.

### 3.3.1 Volumetrisches Ray-Casting

**Ray-Casting** ist eine bekannte Rendering-Methode, die auch beim Rendern von 3D-Szenen zur Anwendung kommt die keine Volumendaten enthalten. In diesem Fall werden von der Position der Kamera für jeden Pixel Strahlen in die Szene geschossen. Für jeden Strahl wird dann errechnet, ob dieser die Oberfläche eines der Objekte in der Szene schneidet. Ist dies der Fall, wird der betreffende Pixel in der Farbe des Objektes eingefärbt, wobei der verwendete Shader diese beeinflusst. Es wird außerdem berücksichtigt, welche Länge der Strahl zu dem Schnittpunkt hat, da nur der Eintrittspunkt relevant ist. Weiterhin existieren die Techniken Ray-Tracing und Ray-Marching, die ebenfalls auf der Kollision zwischen Strahlen und Objekten beruhen. Beim **Ray-Tracing** werden dabei neben dem ursprünglichen Strahl, der das Objekt trifft noch weitere berechnet, die durch die gesamte Szene laufen können um z.B. die Reflexion von Objekten aufeinander zu ermitteln. **Ray-Marching** ist eine schnellere Version des Ray-Castings. Hier wird der Schnittpunkt von Strahl und Oberfläche nicht genau kalkuliert. Stattdessen wird entlang des Strahls in kleiner werdenden Abständen jeweils ein einzelner Punkt betrachtet. Für diesen Punkt wird lediglich geprüft, ob er sich bereits innerhalb des Objektes befindet oder nicht. Der erste Punkt, auf den dies zutrifft wird als Schnittpunkt angesehen. Obwohl diese Berechnung etwas weniger genau ist, ist wie bereits gesagt schneller als Ray-Casting.

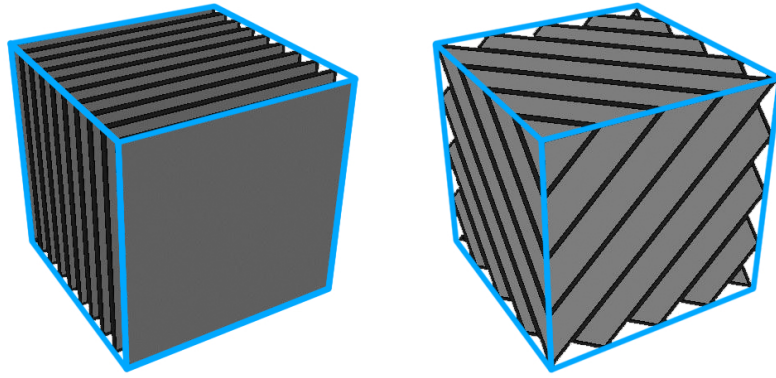


Abbildung 1: Links: Texturen beim werden entlang des Volumens ausgerichtet. Recht: Texturen werden entlang des Blickwinkels ausgerichtet.

Bei beiden Verfahren muss allerdings die Oberfläche bzw. die Form des Objektes bekannt sein, um bestimmen zu können, wann der Strahl in sie eintritt. Dies ist bei Volumendaten in der Regel nicht gegeben, da nicht nur das Objekt selbst, sondern auch dessen Umgebung darin gespeichert sind. Weiterhin werden durch die eben beschriebenen Prozesse nur die Oberflächen der Objekte gerendert, nicht aber ihr Inneres.

Hier liegt der Unterschied zum Rendering von Volumen. In diesem Fall werden beide Verfahren quasi vereint. **Volumetrisches Ray-Casting** und volumetrisches Ray-Marching bezeichnen deshalb ein und dasselbe. Auch hier werden dabei Strahlen von der Kamera in die Szene geschossen. Um das Volumen zu rendern sind davon nur die Strahlen relevant, die das Volumen treffen. Und auch von diesen wird nur der Teil des Strahls berücksichtigt, der innerhalb der umgebenden Geometrie liegt. Für jeden Strahl Ein- und Austrittspunkt errechnet. Innerhalb der Geometrie werden jetzt entlang des Strahls in bestimmten Abständen Voxel betrachtet. Für jeden Voxel wird ein Farb- und Alphawert bestimmt und die Werte werden abhängig von der Komposition von hinten nach vorne oder von vorne nach hinten aufeinander addiert. Die Summen der Werte ergeben so eine dreidimensionale Darstellung des Volumens aus der Sicht der Kamera.

### 3.3.2 Texture Based Volume Rendering

Beim texturbasierten Rendering von Volumen wird eine dreidimensionale Darstellung erzeugt indem viele Texturen aufeinander geschichtet werden, die dann mit Alpha Blending übereinander geblendet werden. Dazu wird für jede Schicht ein Querschnitt durch das Volumen gemacht, auf den dann mit Hilfe von Texture Mapping die Textur gerendert wird. Die Form und Ausrichtung der Querschnitte wird dadurch bestimmt auf welche Art die Volumendaten vorliegen. In Abbildung ?? wird der Unterschied verdeutlicht. Handelt es sich um zweidimensionale Texturen, richten sich die Querschnitte am Volumen aus. Innerhalb eines Würfels sind sie also quadratisch. Damit das Volumen aus verschiedenen Blickwinkeln betrachtet werden kann, wird für jede der Hauptsichtachsen ein Stapel mit Texturen erstellt. Zwischen den Stapeln wird gewechselt, sobald sich der Betrachtungswinkel ändert. Dies ist in Abbildung ?? dargestellt.

Liegen die Daten als 3D-Texturen vor, verlaufen die Querschnitte entlang der Sichtachse. Dazu werden sogenannte Proxy Geometrien erzeugt, Polygone, die einen Querschnitt beschreiben. Auf diesen Proxy Geometrien wird dann die Textur erstellt, indem die entsprechenden Volumendaten abgefragt werden. Dadurch, dass die Querschnitte an der Sichtachse ausgerichtet sind, ist nur ein Texturstapel notwendig.

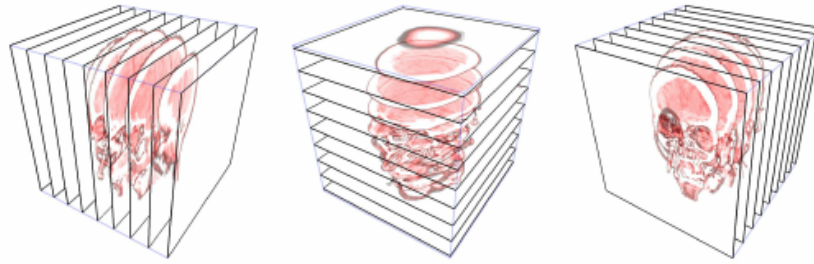


Abbildung 2: Texturen werden als Stapel in der XY-, YZ- und XZ-Ebene angeordnet.

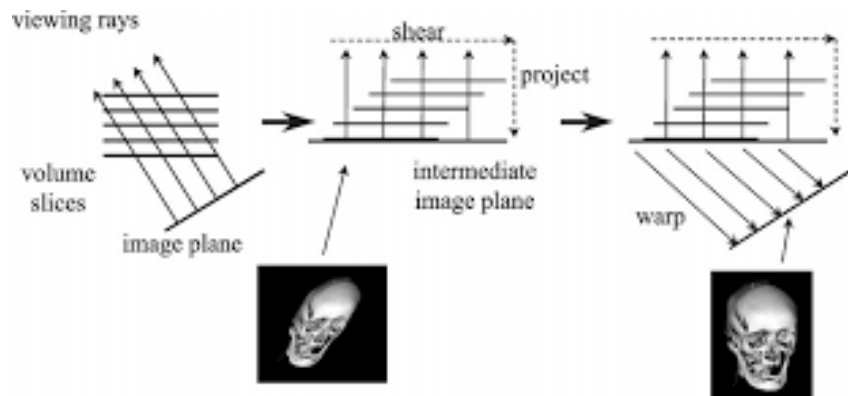


Abbildung 3: Shear-Warp Verfahren. Rechts: Ohne Verschiebung würden die Strahlen der Bildebene schräg auf das Volumen treffen. Mitte: Die Schichten des Volumen werden verschoben, sodass die Strahlen orthogonal darauf treffen. Das resultierende Bild ist verzerrt. Rechts: Das verzerrte Bild wird transformiert, um das Volumen korrekt darzustellen.

<https://dl.acm.org/citation.cfm?id=329138>

### 3.3.3 Shear-Warp

Shear-Warp verfolgt den selben Ansatz wie das Ray-Casting. Anstatt die Strahlen allerdings von der tatsächlichen Kameraposition aus zu verschießen, werden die Strahlen orthogonal zu den Volumenschichten in das Volumen gefeuert. Auf diese Weise wird die Berechnung der Strahlen und die Komposition der Darstellung deutlich beschleunigt. Damit auch bei einem nicht orthogonalen Blickwinkel das Volumen korrekt dargestellt wird, wird vor dem Ray Casting eine vom Blickwinkel abhängige Scherung auf die Volumendaten angewandt. In Abbildung ?? ist Dargestellt, wie die Scherung die einzelnen Schichten entsprechend des Blickwinkels verschiebt, um einen orthogonalen Einfallswinkel zu simulieren. Das durch das Ray Casting entstandene Bild wird zunächst in einen Buffer gerendert. Durch die Scherung ist das Bild zu diesem Zeitpunkt noch verzerrt. Deshalb wird es anschließend noch einmal transformiert, sodass eine korrekte Abbildung des Volumens auf die Bildschirmenebene projiziert wird. In Abbildung ?? ist zu sehen, wie das gerenderte Bild vor und nach der Transformation aussieht.

Shear-Warp vereint damit die Bildqualität des Ray-Castings, ist aber dabei um einiges schneller.

## 3.4 Oberflächen Generierung (Weitere Methoden zur dreidimensionalen Darstellung von MRT-Daten)

Wie in Kapitel 1.1 beschrieben wurde, bietet eine 3D-Darstellung von MRT-Bildern einige Vorteile gegenüber einer Betrachtung der Daten in 2D. Auf Grund dieser Tatsache gab es im Laufe

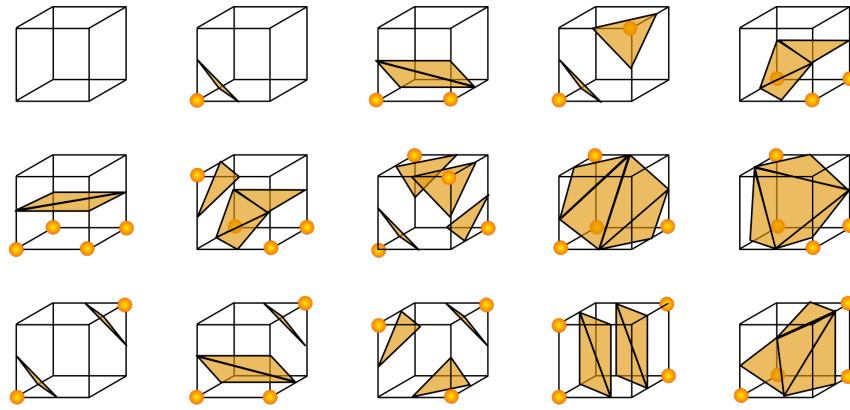


Abbildung 4: Die 15 verschiedenen Möglichkeiten, wie eine Polygonfläche einen Würfel in zwei Bereiche teilen kann.

der Zeit verschiedene Ansätze zur Umsetzung einer dreidimensionalen Darstellungsweise. Im Folgenden werden diese erläutert.

### 3.4.1 Marching Cubes

Das Marching Cubes Verfahren wird eingesetzt, um eine Polygonenoberfläche zu erzeugen, die ein volumetrisches Objekt visualisiert. Der Visualisierung liegen Volumendaten, die in Form eines Skalarfeldes vorliegen zu Grunde. Die Idee hinter dem Algorithmus ist, dass es zwischen den Voxeln, die innerhalb und außerhalb eines Objektes liegen eine Grenze gibt, die die Oberfläche des Objektes bildet. Diese Grenze gilt es zu finden. Die Werte der Voxel, werden dabei als Dichtewerte angesehen. Voxel innerhalb der Grenze haben eine Dichte, die außerhalb eine andere. Demnach kann ein Schwellenwert definiert werden, der die beiden Wertebereiche voneinander trennt. Um die Grenze zwischen den Voxeln zu ermitteln, werden diese zunächst in Würfel aufgeteilt, woher auch der Name des Verfahrens stammt. Immer vier benachbarte Voxel bilden dabei einen Würfel.

Die Vorstellung einer Grenze zwischen den Voxeln lässt sich auf die Würfel übertragen. Dabei liegt dann ein bestimmter Teil der Würfel des Volumens mit allen Eckpunkten innerhalb und ein bestimmter Teil außerhalb des Objektes. Diese Würfel liegen nicht auf der Grenze und alle ihre Eckpunkte haben jeweils ähnliche Werte. Andererseits gibt es Würfel, durch die die Grenze verläuft. Die Eckpunkte dieser Würfel können sich in ihren Werten stark unterscheiden, da jeweils ein Punkt zum Inneren und ein anderer zum Äußeren des Objektes gehören kann.

Um also die Oberfläche zu erzeugen, die das gesamte Volumen aufteilt, wird jeder Würfel zuerst einzeln betrachtet und es wird ermittelt, ob und wie die Polygonfläche diesen einzelnen Würfel aufteilt. Da diese Teilung auch einzelne Eckpunkte betreffen kann, setzt sich diese Oberfläche aus dreieckigen Polygonen zusammen. ?

Die Anzahl an Möglichkeiten einen Würfel aufzuteilen lässt sich errechnen. Für jeden der acht Eckpunkte des Würfels gilt eine von zwei Möglichkeiten: Entweder befindet er sich unter der Oberfläche und damit im Objekt oder eben nicht. Demnach ergeben sich für alle Punkte zusammen  $2^8 = 256$  mögliche Fälle der Aufteilung. Durch in Betracht ziehen von Symmetrie, könnten diese auf 15 verschiedene Möglichkeiten reduziert werden, die in Abbildung 1 dargestellt sind. Der Fall, dass die Eckpunkte gar nicht getrennt werden ist dabei berücksichtigt.

Bei der Implementierung des Algorithmus werden nun alle 256 Teilungsmöglichkeiten eines Würfels in einer Look-up-Tabelle gespeichert. Um auszulesen, welcher Fall auf den momentan betrachteten Würfel zutrifft, bedarf es eines Indexes. Der Index ist eine achtstellige Binärzahl. Sie wird ermittelt, indem für jeden der acht Eckpunkte des Würfels geprüft wird, ob er unter



dem Schwellenwert liegt, der die Unterscheidung zwischen Innen und Außen definiert oder nicht. Die Ergebnisse der Prüfung, die durch 1 oder 0 repräsentiert werden, werden aneinander gefügt. Nach der Prüfung aller Eckpunkte ist so ein achtstelliger Index entstanden.

Durch die Verwendung dieses Index und der Tabelle lässt sich so auslesen, wie die Polygone in dem Würfel liegen. Damit ist auch bekannt, auf welchen Würfelkanten sich deren Vertices befinden. Für die Erzeugung einer Oberfläche muss weiterhin die genaue Position des Vertex auf der jeweiligen Kante ermittelt werden. Dies geschieht durch lineare Interpolation zwischen den Werten der Würfel-Eckpunkte, zwischen denen die betreffende Kante verläuft. ?? Auf die selbe Weise wird für jeden Vertex der Polygone dessen Normale berechnet, indem die Normalen der Eckpunkte interpoliert werden.

Nach und nach werden so für jeden Voxel-Würfel Polygone berechnet, die in einer Liste gespeichert und schließlich gezeichnet werden.

Der Marching Cubes Algorithmus wird häufig zur prozeduralen Erzeugung von Terrain eingesetzt. In diesem Fall ist es entscheidend, wie die Dichtewerte des Volumens generiert werden. Bei der Visualisierung von medizinischen Daten, wie MRT-Bildern werden die Bilddaten als Eingabewerte genommen.

Beschleunigung durch Unterstützung der Grafikkarte??

Erweiterungen/ Abwandlungen

### 3.4.2 Voxel

### 3.4.3 Andere

## 3.5 Vergleich der Methoden im Überblick

## 3.6 AR und VR (MR?)

Occluded Display

### 3.6.1 Augmented Reality

Der Begriff der *Augmented Reality* (AR), also *Erweiterte Realität* beschreibt die Idee, dass die physisch vorhandene Welt, die einen Nutzer umgibt angereichert wird mit zusätzlichen digitalen Inhalten. Dies geschieht z.B. indem wahrgenommene Gegenstände von virtuell erzeugten Objekten überlagert werden. Laut [azuma97] zeichnet sich eine AR-Anwendung durch folgende Eigenschaften aus:

- Kombiniert reale und virtuelle Realität
- Ist in Echtzeit interaktiv
- 3D-Objekte existieren in dreidimensionaler Realität

AR und VR verbindet eine ähnliche Geschichte. Die frühesten VR-Systeme waren nach heutiger Definition AR-Systeme. Auch einige AR-HMDs funktionieren mit Stereoskopie. Die eben genannten Definition lässt allerdings auch andere Systeme zu. Tatsächlich gibt es eine Vielzahl von AR-Systemen. Beispielsweise kann eine AR-Anwendung mit oder manche ohne einen Bildschirm konzipiert werden, der die Augmentierung darstellt. Im letzteren Fall werden dann meist Beamer o.Ä. verwendet, um die Objekte in die reale Welt zu projizieren. Die Mehrzahl von AR-Software verwendet wahrscheinlich allerdings Bildschirme. Aber auch hier gibt es verschiedene Lösungsansätze. Dies betrifft zum einen die Positionierung des Bildschirms. Dieser kann direkt vor dem Auge des Nutzers, in der Welt oder in einem tragbaren Endgerät platziert werden. Weiterhin können Bildschirme durchsichtig oder undurchsichtig sein. Handelt es

sich um einen durchsichtigen Bildschirm, wird darauf nur die Augmentierung dargestellt. Des Vorteil hierbei ist, dass der seine Umgebung größtenteils unverfälscht wahrnehmen kann. Damit ist aber gleichzeitig der Kontrast zu den nicht unbedingt realitätsnahen 3D-Objekten größer. Bei undurchsichtigen Bildschirmen nimmt eine Kamera die Umgebung auf und gibt sie augmentiert auf dem Bildschirm aus. Diese Technik wird z.B. verwendet wenn AR-Anwendungen für Smartphones oder Tablets entwickelt werden. In Abbildung ?? ist dargestellt, wie eine solche Anwendung auf einem Smartphone aussieht.

Alle Techniken haben gemeinsam, dass eine Kamera Teil des Systems ist, die die Umgebung des Nutzers aufnimmt. Dadurch werden Tiefen- und Bildinformationen gesammelt. Durch die Auswertung dieser und anderer Informationen, wie z.B. GPS- oder Wlan-Daten, können 3D-Objekte korrekt mit der Umgebung in Beziehung gesetzt werden. So ist es möglich ein virtuelles Objekt auf oder hinter ein reales zu stellen. Durch ein möglichst umfassendes Verständnis des umgebenden Raumes ist es auch möglich, dass die Elemente der erweiterten Realität in der realen Welt an der richtigen Stelle platziert werden und dort auch bleiben. Alternativ dazu können am gewünschten Ort Markierungen gesetzt werden, die das System erkennt und somit genau dort das 3D-Modell darstellen kann. Solche Marker können verschieden aussehen. Oft werden komplexe weiße Muster dafür verwendet, weil diese gut zu identifizieren und schwer zu verwechseln sind. In Abbildung ?? ist ein 3D-Objekt zu sehen, dass auf einem Marker platziert wurde.

Zurzeit gibt es verschiedene AR-Systeme auf dem Markt, die sich teilweise in ihrer Funktionsweise unterscheiden. Die wichtigsten sind hier aufgezählt.

Smartphones. Wie bereits erwähnt können Smartphones oder Tablets als Fenster in die erweiterte Realität dienen. Dazu kann wahrscheinlich jedes Smartphone mit Kamera genutzt werden. Es gibt verschiedene SDKs und Frameworks zur Entwicklung von AR-Apps. Dazu gehören unter anderem *ARKit* von Apple, *ARCore* von Google und *Vuforia*.

Binoculare HMDs. Hierzu zählen HMDs, die vor jedem Auge des Nutzers einen Bildschirm platzieren und so stereoskopisch dreidimensionale Inhalte simulieren. Im AR-Bereich sind dabei die *Hololens* von Microsoft sowie das Headset von Magic Leap die wichtigsten Vertreter der Technologie. Beide Geräte haben durchsichtige Bildschirme und können unabhängig von einem Rechner verwendet werden. An dieser Stelle sollen auch VR-Geräte, wie die HTC Vive Pro genannt werden, die durch ihre eingebaute Kamera auch die Umgebung auf ihren Bildschirmen darstellen kann. Diese kann dabei natürlich auch mit virtuellen Inhalten erweitert werden.

Ebenfalls als HMD zu bezeichnen ist die *Google Glass* von Google. Dabei handelt es sich um ein Brillengestell, dass vor einem der Augen des Nutzers einen durchsichtigen Bildschirm fixiert, auf dem digitale Inhalte dargestellt werden können. Google Glass wird momentan allerdings nur noch in großen Firmen verwendet.

### 3.6.2 Virtual Reality

Wenn die Realität durch AR also erweitert wird, so wird sie durch Virtual Reality völlig ersetzt. D.h. der Nutzer wird von seiner realen Umgebung abgeschnitten und in eine simulierte versetzt, mit der er in Echtzeit interagieren kann. Dabei wird in der Regel eine möglichst hohe Immersion angestrebt. Im Kontext von VR bezeichnet Immersion den Effekt, dass ein Nutzer dermaßen in die simulierte Welt, die umgibt eintaucht, dass sich diese für ihn real anfühlt und Interaktionen mit ihr natürlich werden. Wie immersiv eine Anwendung ist hängt dabei von der Anwendung selbst ab, sowie auch von dem verwendeten VR-System.

Im Laufe der Zeit wurden verschiedene Systeme entwickelt, die eine möglichst immersive Realität erschaffen sollten. Bereits in den 50er Jahren wurde versucht den Nutzer in das Geschehen eines Films zu versetzen, indem er von seiner realen Umgebung isoliert wurde, und seine Wahr-

nehmung so auf bestimmte Reize gebündelt wurde. In den 1960ern wurden dann erste Versionen eines Head Mounted Displays (HMD) entwickelt. Dabei handelt es sich um eine Art Brille, mit zwei Bildschirmen vor den Augen, auf die simulierte Inhalte projiziert werden. Um den Eindruck von Dreidimensionalität zu erzeugen, wird dabei das Prinzip der Stereoskopie verwendet. Hierzu werden jeweils dem linken und rechten Auge des Nutzers zwei Bilder des selben Objektes aus leicht unterschiedlichen Blickwinkeln gezeigt. Da die Augen beim normalen Sehen ein Objekt tatsächlich aus zwei verschiedenen Winkeln wahrnehmen, werden die Bilder im Gehirn zusammengefügt, sodass der Eindruck von Tiefe entsteht. Die Bilder werden dabei auf zwei Bildschirmen direkt vor dem Auge angezeigt, vor die eine Linse gesetzt wird, um ??

Der Blickwinkel auf ein Objekt hängt dabei von der Kopfposition des Nutzers ab. Um diesem die Möglichkeit zu geben, den Kopf zu bewegen und damit den Eindruck eines 3D Objektes zu verstärken wird die Position des Kopfes getrackt. Auf diese Weise können die stereoskopischen Bilder dem Blickwinkel des Nutzers angepasst werden. Dieser erhält somit das Gefühl, er könne um das dargestellte Objekt herumgehen.

Die Bewegungsfreiheit der Nutzer war allerdings durch die damalige Technik eingeschränkt, denn das System, mit dem das HMD verbunden war, war zu schwer, um es zu tragen und deshalb fest installiert. In den 90er Jahren gab es deshalb Entwicklungen in eine andere Richtung. Anstatt die virtuelle Realität nur direkt vor den Augen des Nutzers darzustellen, sollte diese um ihn herum erzeugt werden. Dazu wird der Nutzer in einem abgeschlossenen Raum platziert, an dessen Wände und Boden die Umgebung des jeweiligen Szenarios projiziert wurde. Die Immersion des Erlebnisses kann gesteigert werden, indem z.B. 3D-Brillen eingesetzt werden. Sogenannte CAVE-Systeme (Cave Automatic Virtual Environment) kommen auch heute noch zum Einsatz. Da sie allerdings sehr unhandlich und in ihrer Umsetzung kostspielig sind, eignen sie sich nicht als Massenprodukt oder für private Nutzung.

Obwohl die Simulation einer virtuellen Realität oft auf die Täuschung visueller Wahrnehmung fokussiert ist, werden in vielen Systemen auch andere Sinne angesprochen, um das Erlebnis immersiver zu gestalten. Dies gilt vor allem für das Gehör. Dem Nutzer werden dabei über Lautsprecher oder Kopfhörer zur Erfahrung passende Geräusche vorgespielt. Eine größere Herausforderung stellt die Imitation von haptischen Reizen dar. Hierzu gibt es verschiedene Ansätze. Einer der einfacheren ist es dem Nutzer bei virtuellen Berührungen Vibrationen auszusetzen, z.B. über Eingabe Medien in seinen Händen. Es gibt allerdings mehrere Konzepte, die Haptische Impulse realistischer simulieren sollen, beispielsweise durch Handschuhe aus speziellem Material. Durch die Verwendung eines Laufbands, das auf die Laufbewegung des Nutzers reagiert soll die Bewegungsfreiheit innerhalb der Simulation erweitert werden. Omnidirektionale Laufbänder können sowohl zusammen mit HMD als auch CAVEs eingesetzt werden. Über den Verlauf der Entwicklung von VR-Systemen wurde außerdem des Öfteren versucht Gerüche zu simulieren.

Auf die Möglichkeiten der Nutzerinteraktion in VR-Systemen wird im Abschnitt 3.8 genauer eingegangen.

Heutzutage bestehen die meistgenutzten VR-Systeme meist aus einem HMD, welches über eine Kamera verfügt, Eingabemedien (z.B. Controllern), einem Trackingsystem, das die Position der ersten beiden Komponenten verfolgt und einem Computer, der die Komponenten miteinander verknüpft und auf dem die VR-Anwendung läuft.

Im Folgenden sind Hersteller von VR-Systemen aufgelistet, sowie die Modelle, die derzeit auf dem Markt zum freien Verkauf stehen.

- Facebook (Oculus Rift und Oculus Go)
- Google (Google Cardboard, Google Daydream)
- HTC & Valve (HTC Vive)

- Microsoft (Microsoft HoloLens, Windows Mixed Reality)
- Razer (OSVR Hacker Dev Kit)
- Samsung (Samsung Gear VR)
- Sony Computer Entertainment (PS VR)
- Starbreeze Studios (StarVR)
- Lenovo (Lenovo Explorer für Windows Mixed Reality und Lenovo Mirage Solo für Google Daydream)

### 3.7 Verarbeitung von MRT-Daten

Obwohl die Darstellung von MRT-Bildern in direktem Zusammenhang mit ihrem Zweck steht, bleibt die Verarbeitung der Daten auch bei verschiedenen Darstellungsformen gleich.

#### 3.7.1 Arbeit eines Neurologen

Wie bereits in Kapitel 1.1 oberflächlich erläutert wurde, besteht der Nutzen eines MRTs darin, dass der behandelnde Arzt einen Einblick in die betreffenden Organe (hier das Gehirn) erhält. Dazu studiert er die einzelnen Schichten des Gehirns und hält Ausschau nach Anomalien. Im Fall von Schlaganfällen sind diese ...

Wurde ein entsprechender Bereich identifiziert, markiert der Arzt diesen auf jeder einzelnen Schicht.

#### Schlaganfälle auf einem MRT

#### 3.7.2 Software in der Radiologie

Die Untersuchung der MRT-Daten werden digital durchgeführt. Um dem Arzt einen Einblick in den Datensatz zu geben, gibt es spezielle Software, die diesen darstellen kann und weiterhin relevante Funktionen bietet. Beispiele für solche Software sind die Programme

Drei Achsen (namen?) verschiedene Sequenzen? Interaktionen

Seit es die Technologie erlaubt (?) erfolgt die Verarbeitung von MRT-Daten digital. Dazu gehört auch, dass der Arzt die Bilder genau studiert, um eine Diagnose stellen zu können. Wie bereits beschrieben, liegen die MRT-Bilder meist im DICOM oder nifti Datenformat vor. Um die Dateien öffnen zu können, bedarf es bestimmter Software. Diese kann sich von Arzt zu Arzt unterscheiden. In Krankenhäusern werden die Bilder allerdings in der Regel in einem Picture Archiving and Communication System (PACS) gespeichert. (Fußnote?) Dabei handelt es sich um einen Server, der unter anderem medizinische Bilddaten zentral speichert. Die Bilder werden von bildgebenden Verfahren aus z.B. der Radiologie direkt im PACS gespeichert. Von dort aus kann entweder mit speziellen Arbeitsplatzrechnern oder auch mit herkömmlichen Computern über den Browser auch die Bilder zugegriffen werden. Dies geschieht dann meistens durch einen entsprechenden Viewer, der in das PACS integriert ist. ??

Diese PACS-Viewer unterscheiden sich oft nicht stark von anderen DICOM Viewern, von denen eine Vielzahl gibt. Teilweise lassen sich kostenlose Viewer im Internet finden.

Vom Aufbau her sind diese wie gesagt alle recht ähnlich. Wie im Kapitel 1.1 bereits oberflächlich beschrieben, stellen sie das betreffende Organ, z.B. ein Gehirn, aus drei verschiedenen Blickwinkeln dar, die an den X-, Y- und Z-Achsen ausgerichtet sind. Orthogonal zum Blickachse steht jeweils eine Ebene, die gleichzeitig den Querschnitt durch das Gehirn darstellt. Diese Ebenen können entlang der Achsen verschoben werden. Auf diese Weise scrollt der Neurologe durch die

verschiedenen Schichten des Gehirns. Bei den drei Ebenen handelt es sich um die Frontal- oder Coronelebene, die das Gehirn in vorne und hinten teilt, die Sagittalebene, die zwischen links und rechts verläuft und die Transversalebene, die eine Teilung zwischen oben und unten bewirkt. Die drei Ebenen sind dabei in jeder Ansicht farbkodiert eingezeichnet, sodass der Arzt ein besseres Bild davon bekommt, welchen Teil des Gehirns er gerade betrachtet. Manchmal wird noch eine vierte Ansicht ergänzt, auf der entweder die Sagittalebene von der anderen Seite abgebildet ist oder sogar durch das Ineinanderschieben der gerade angezeigten Schichten eine annähernd dreidimensionale Darstellung simuliert wird. In Abbildung ?? ist die Oberfläche der XY-Viewers zu sehen, die beispielhaft die Ansichten zeigt.

Die Benutzeroberfläche bietet auch einige Interaktionen. Die wichtigsten sind dabei die, die es dem Arzt ermöglichen ein möglichst eindeutiges Verständnis vom Inneren des Gehirns zu erhalten. Dazu zählen:

#### **Scrollen durch die einzelnen Bildschichten**

Dies geschieht in der Regel durch die Auswahl einer der Ansichten und die Betätigung des Mausekaders.

#### **Einstellen von Kontrast und Helligkeit**

Dadurch können schlecht sichtbare Strukturen erkennbarer gemacht werden. Das Werkzeug dazu muss in der Taskleiste ausgewählt werden. In einem separaten Fenster können die Werte dann angepasst werden. ??

#### **Heranzoomen**

Bestimmte Bereiche eines Bildes können vergrößert werden, um sie besser beurteilen zu können.

#### **Verschieben des Bildausschnitts**

Um über ein vergrößertes Bild zu navigieren, kann der Nutzer dieses anklicken und in eine Richtung ziehen. Dies ermöglicht es andere Bildteile zu untersuchen, ohne zuerst herauszoomen zu müssen.

Obwohl über die Taskleiste oft noch weitere Optionen zur Verfügung stehen sind diese nicht essenziell zur Untersuchung der Bilder und werden meistens nicht oder nur in geringem Umfang genutzt.

### **3.8 Interaktion in AR/VR**

#### **3.8.1 Systeminterne Nutzereingaben**

VR und AR Systeme, die es heutzutage zu kaufen gibt besitzen in der Regel eine oder mehrere Formen der Eingabe, um mit dem System zu interagieren.

Die Eingabemöglichkeiten sind dabei teilweise von der Art des Systems abhängig. Beispielsweise kann für Smartphone-Apps der Touchbildschirm des Gerätes für die Nutzereingabe genutzt werden.

Theoretisch kann jedes System mit jedem Eingabemedium verknüpft werden. Dies ist allerdings für Nutzer und Entwickler mit entsprechendem Aufwand verbunden. Die Techniken, die von den derzeit erhältlichen AR und VR Systemen verwendet werden, um Nutzereingaben zu erfassen lassen sich in zwei Kategorien teilen.

**Controller** Zum einen gibt es Controller, die mit dem System zusammen hergestellt werden und die der Nutzer während der Verwendung des Systems in der Hand hält. Im Fall von VR Systemen sind meistens zwei Controller vorhanden, einer für jede Hand. Diese sind jeweils mit einer Anzahl an Knöpfen oder auch Touchpads versehen, die der Nutzer bedienen kann. Im

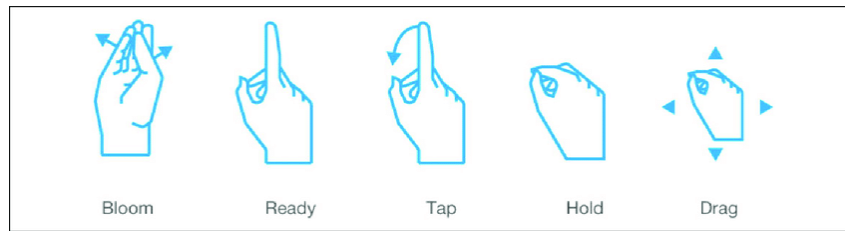


Abbildung 5: Handgesten, die die Hololens erkennt.

#### Hand Gestures

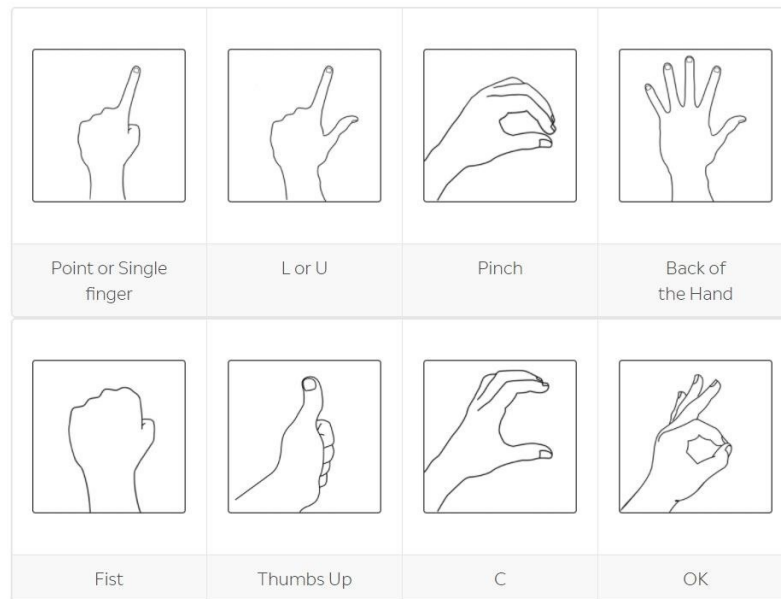


Abbildung 6: Handgesten, die die Magic Leap erkennt.

Aussehen sind sich Controller dieser Art recht ähnlich. In Abbildung ?? ist der Controller des HTC Vive Systems zusehen. Allerdings existieren auch Systeme für dessen Bedienung nur ein Controller vorgesehen ist. Ein VR-Beispiel ist die *Oculu Go*, ein eigenständiges System für grafisch weniger anspruchsvolle Anwendungen. Auch AR-Systeme, wie die *Magic Leap* oder die *Hololens* besitzen nur einen Controller. Diese haben im Vergleich zu den zuerst genannten VR-Controllern deutlich weniger Eingabemöglichkeiten. Im Falle der *Hololens* handelt es sich um einen flachen Controller, den sich der Nutzer an den Finger steckt. Indem er Druck ausübt, kann er klicken. Daneben wird nur die Position und Rotation des Clickers verfolgt, durch die der Nutzer mit dem System interagieren kann. Das Gerät stellt damit eine von zwei möglichen Eingabemethoden dar.

**Gesten** Die andere ist die Steuerung über Gesten, die vor allem AR-Systeme verwenden und die zweite Kategorie darstellt. Die in den Systemen verbaute Kamera erfasst dabei die Hände des Nutzers in ihrem Blickfeld und erkennt bestimmte Handgesten, auf die dann reagiert wird. Die *Hololens* kann zum aktuellen Zeitpunkt zwei Hauptgesten erkennen, *Bloom* und *Air tap*, sowie die Bewegung der Hand. Durch die Verbindung von beidem können beispielsweise Objekte verschoben werden. In Abbildung ?? ist dargestellt, wie die Hololens-Gesten ausgeführt werden. Die *Magic Leap* erkennt dagegen acht verschiedene Handgesten, die in ?? abgebildet sind.

#### Kopfposition und Blickrichtung

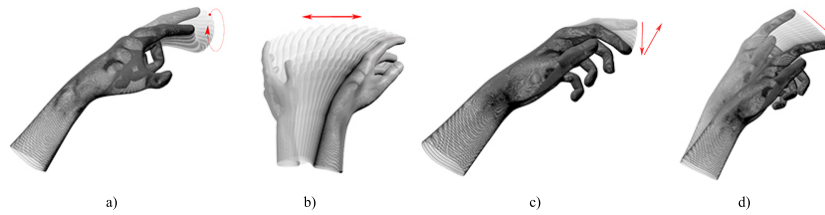


Abbildung 7: Handgesten, die die Leap Motion erkennt. *Circle* (a), *Swipe* (b), *Tap* (c) und *Screen Tap* (d).

## Sprachsteuerung

**HMD-Knöpfe** Schließlich besitzen in vielen Fällen auch die HMDs der Systeme Knöpfe oder Räder. Diese dienen allerdings meistens zur Konfiguration des Systems und Steuern z.B. die Lautstärke.

Controller

### 3.8.2 Leap Motion

Die *Leap Motion* ist ein externes Gerät, dass die Hände des Nutzers tracked und die genauen Handbewegungen an einen Computer übermitteln kann. Dies funktioniert mittels Tiefeninformationen, die die in *Leap Motion* eingebaute Kamera sammelt. Das Gerät wird über USB an einen Rechner angeschlossen und kann dann als unabhängiges Eingabemedium verwendet werden. Allerdings ist auch eine Verwendung in Kombination mit AR- oder VR-Systemen möglich. Die *Leap Motion* schafft es die Handbewegungen des Nutzers verlässlich anzufinden, sodass die Hände in den digitalen Raum übertragen werden können. Hier kann nun eine Interaktion mit anderen digitalen Objekten stattfinden.

Die *Leap Motion* erkennt weiterhin vier Handgesten. Diese sind *Circle*, *Swipe*, *Tap* und *Screen Tap*. Die Gesten sind in Abbildung ?? zu sehen.

### 3.8.3 Andere

Consolen Controller

### 3.8.4 Interaktionsdesign

## Kapitel 4

### Anforderungsanalyse

#### 4.1 Interviews

Zur Bestimmung der Anforderungen, die von der Anwendung erfüllt werden sollen wurden iterativ mehrere Interviews mit dem betreuenden Neurologen der Charité geführt.

Wie bereits beschrieben, ist es das Ziel dieser Arbeit Neurologen eine dreidimensionale Darstellung von Gehirn-MRT-Scans in einer interaktiven AR- bzw. VR-Anwendung zur Verfügung zu stellen, um zu untersuchen, inwiefern diese die Ärzte in ihrer Arbeit besser unterstützt, als bisher verwendete Darstellungssoftware. Die hypothetischen Vorteile eines solchen Programmes wurden im Kapitel 1.1 erläutert. Für den Vergleich mit bisheriger Software ist eine Anwendung prototypischen Charakters ausreichend. Diese soll auf die grundlegenden Interaktionen beschränkt sein, die ein Neurologe für die Verarbeitung der Daten benötigt.

Der Fokus der Untersuchung liegt auf dem Bereich der akuten Schlaganfall Behandlung. Wie im Kapitel ?? beschrieben untersucht der Neurologe hierfür die Abbildungen des Gehirns eines Patienten, die durch den MRT-Scan erzeugt wurden. Dabei hält er nach Bereichen Ausschau, die von einem Schlaganfall betroffen sein könnten, um die Art des Schlaganfalls zu bestimmen, ? sowie den entstandenen Schaden einschätzen zu können. Betroffene Bereiche werden anschließend markiert. Die Markierung kann danach ein- und ausgeblendet werden.

Weiterhin könnte man anhand des Modells auch Voraussagen von Therapieerfolgen anschaulich demonstrieren.

Folgende Anforderungen haben sich aus den Interviews des Neurologen ergeben, die erfüllt sein sollten, um den erfolgreichen Einsatz von mARt zu ermöglichen.

##### 4.1.1 Anforderungen aus Interviews

Die aus dem Interview ermittelten Anforderungen wurden in der folgenden Tabelle aufgelistet. Zur besseren Referenzierung wurden sie jeweils mit Bezeichner versehen. Weiterhin wurde die Priorität jeder Anforderungen auf niedrig, mittel, oder hoch geschätzt.



Anforderungsbezeichnung	Anforderung	Priorität
A01	3D Darstellung der MRT Daten	hoch
A02	Einblenden des gekennzeichneten Bereichs in 3D	hoch
A03	Scrollen durch Schichten der 3D-Darstellung auf mindestens einer Achse	mittel
A04	Erkennbarkeit der inneren Struktur des Gehirns (entsprechend der MRT-Bilder)	hoch
A05	Scrollbare 2D Darstellung der MRT-Bilder auf mindestens einer Achse	mittel
A06	Kontrast und Helligkeit sollten einstellbar sein	hoch
A07	Man sollte die Darstellung vergrößern können	hoch
A08	Die Darstellung sollte verschiebbar sein	hoch
A09	Interaktionselemente sollten die Darstellung nicht verdecken	niedrig
A10	Scrollen durch Verwendung eines Scrollrads	niedrig
A11	Auswahl verschiedener MRT-Sequenzen (falls vorhanden)	niedrig
A12	Unterstützung von nifti-Daten	mittel
A13	Verschiedene Ansichten können nebeneinander dargestellt werden. (z.B. vor und nach Therapie)	mittel

Tabelle 4.1: Durch Interview bestimmte Anforderungen.

## 4.2 User Stories

Anhand dieser Anforderungen wurden User Stories formuliert.

Story-Bezeichnung	User Story	Anforderungsreferenz
U01	Als Nutzer möchte ich ein 3D-Modell des gescannten Gehirns sehen, um ...	A01
U02	Als Nutzer möchte ich den gekennzeichneten Bereich innerhalb des Gehirn sehen können, um einzuschätzen wie groß der Bereich tatsächlich ist.	A02
U03	Als Nutzer möchte ich den gekennzeichneten Bereich ein- und ausblenden können, um mich auf diesen, oder die Darstellung an sich konzentrieren zu können.	A02
U04	Als Nutzer möchte ich auch dann noch die Strukturen des Gehirns erkennen, wenn der gekennzeichnete Bereich eingeblendet ist.	A03
U05	Als Nutzer möchte ich eine möglichst genaue und gut erkennbare Abbildung der MRT-Bilder, damit ich eine Vorstellung habe, wie das Gehirn des Patienten aussieht.	A05
U06	Als Nutzer möchte ich durch das 3D-Modell scrollen können, um es vergleichbar mit der 2D Darstellung zu verwenden.	A04
U07	Als Nutzer möchte ich die MRT-Darstellungen frei im Raum bewegen können, um sie meiner Position im Raum anzupassen.	A07
U08	Als Nutzer möchte ich die MRT-Darstellungen skalieren können, um die Darstellung gut zu erkennen.	A08
U09	Als Nutzer möchte ich die MRT-Darstellungen drehen können, um den besten Blickwinkel auf den für mich relevanten Bereich zu bekommen.	A09
U10	Als Nutzer will ich, dass die Darstellung nicht von anderen Elementen verdeckt wird, damit ich sie uneingeschränkt sehen kann.	A10
U11	Als Nutzer möchte ich mit einem Scrollrad durch die Darstellung scrollen können, um genaue Kontrolle darüber zu haben, welche Schichten angezeigt werden.	A11
U12	Als Nutzer möchte ich alle vorhandenen MRT-Sequenzen sehen und zwischen ihnen wählen können, damit ich alle notwendigen Informationen zu dem Scan nutzen kann.	A12
U13	Als Nutzer möchte ich Dateien im nifti-Format in der Anwendung verwenden können, damit ich sie nicht vorher umwandeln muss.	A13
U14	Als Nutzer möchte ich aus verschiedenen Darstellungen wählen können, die nebeneinander angezeigt werden, um direkte Vergleiche zwischen diesen ziehen zu können.	A14

Tabelle 4.2: Aus Anforderungen abgeleitete User Stories.

### 4.3 Technische Anforderungen

## Kapitel 5

### Konzept

Im folgenden werden Konzepte diskutiert, um die in Kapitel 4 herausgearbeiteten Anforderungen zu erfüllen.

#### 5.1 3D Darstellung

- Da das innere der 3D Darstellung erkennbar sein soll, muss eine semi-transparente Darstellung erzeugt werden, die sowohl die Form des Gehirns abbildet, als auch die innere Struktur - Relevant sind nur die Pixel, die das Gehirn darstellen. Der Bereich darum (Schädel und Hintergrund) muss gefiltert werden. Das gewünschte Ergebniss wäre, wenn das Gehirn frei im Raum "schwebt" (Bei Ray Casting noch mal erwähnen?)

- Daten liegen dreidimensional in Schichten vor - verschiedene Möglichkeiten zur 3D Visualisierung - - Voxel - - Volume Rendering (Ray Casting) - - Marching Cubes - - ...?

- Ist es notwendig bzw. nützlich ein Mesh zu generieren? - Gut für Darstellung der Gehirnform. Relevant ist aber vor allem das "Innere" des Gehirns, da sich dort der markierte Bereich befindet.

- Marching Cubes eignen sich nicht, um innere Struktur darzustellen. Eine Transparente Ansicht würde das Modell als innen "gleich" darstellen - Das Mesh müsste kontinuierlich angepasst werden, um jeweils andere Schichten beim scrollen anzuzeigen.

- Voxel könnten auch das Innere des Gehirns darstellen. Um ein deutliches und hochaufgelöstes Modell zu erhalten wären allerdings viele Voxel notwendig. Performance? - Referenzen

- Im Fokus der Darstellung, soll das Innere des Gehirns stehen. Ein Mesh, das die äußere Form beschreibt ist also nicht sinnvoll, zumal keine der Anforderungen Funktionen beschreibt, für die ein Mesh nötig wäre (z.B. Kollision (Unity))

Wie in dem Kapitel 2 beschrieben, gibt es bereits viele Lösungsansätze zur 3D Darstellung von MRT-Bildern. Obwohl verschiedene Methoden zum Einsatz kommen, ist der am weitesten verbreitete Ansatz der des Volume Rendering. Dies hat die folgenden Gründe: - - ...

Diese Vorgehensweise ist somit am besten für die Implementierung der Anwendung geeignet. Diese wird im Kapitel 6 genauer beschrieben.

Anforderungen an Shader??

#### 5.2 Darstellung des gekennzeichneten Bereichs

U02 U03

#### 5.3 Endgerät

- Hololens oder HTC Vive? - Leistung - Interaktionsmöglichkeiten - Tragekomfort - ...

## 5.4 Interaktion/ UI

### 5.4.1 Interaktion AR

### 5.4.2 Interaktion VR

U06 Als Nutzer möchte ich durch das 3D-Modell scrollen können, um es vergleichbar mit der 2D Darstellung zu verwenden.

U07 Als Nutzer möchte ich die MRT-Darstellungen frei im Raum bewegen können, um sie meiner Position im Raum anzupassen.

U08 Als Nutzer möchte ich die MRT-Darstellungen skalieren können, um die Darstellung gut zu erkennen.

U09 Als Nutzer möchte ich die MRT-Darstellungen drehen können, um den besten Blickwinkel auf den für mich relevanten Bereich zu bekommen.

U10 Als Nutzer will ich, dass die Darstellung nicht von anderen Elementen verdeckt wird, damit ich sie uneingeschränkt sehen kann.

U11 Als Nutzer möchte ich mit einem Scrollrad durch die Darstellung scrollen können, um genaue Kontrolle darüber zu haben, welche Schichten angezeigt werden.

U12 Als Nutzer möchte ich alle vorhandenen MRT-Sequenzen sehen und zwischen ihnen wählen können, damit ich alle notwendigen Informationen zu dem Scan nutzen kann.

U14 Als Nutzer möchte ich aus verschiedenen Darstellungen wählen können, die nebeneinander angezeigt werden, um direkte Vergleiche zwischen diesen ziehen zu können.

## 5.5 Unterstützung von Dateiformaten

Die User Story U13 verlangt nach einer Unterstützung des nifti-Dateiformats.

nifti-Dateien, sind Bilddateien, die oft in der Medizin verwendet werden. Sie speichern Bildsequenzen und eignen sich somit für MRT-Bilder. Ein weiteres für diesen Zweck weitverbreitetes Dateiformat ist DICOM. ...

Es ist weiterhin sinnvoll gängigere Dateiformate, wie JPEG oder PNG zu unterstützen.

Wie in 4 beschrieben, soll es sich bei mArt in erster Linie um einen Prototyp handeln. Der Datensatz, der dargestellt werden soll ist gering. Um die Komplexität und den Umfang der Anwendung möglichst gering zu halten kann darauf verzichtet werden, dem Nutzer die Möglichkeit zu geben, aus verschiedenen Dateitypen oder Datensätzen zu wählen. Es ist ausreichend, wenn der entsprechende Datensatz vor dem Build gewählt wird.

Da die Darstellung und Interaktion eines Datensatzes also im Vordergrund stehen soll, ist es sinnvoll, die Dateien in ein Format umzuwandeln, das einfacher zu verarbeiten ist (JPEG, PNG). Hierzu kann ein externes Tool verwendet werden (ImageJ Bezug in Implementierung).

## Kapitel 6

### Implementierung

#### 6.1 Aufbau Struktur des Projektes

##### 6.1.1 Kombination von Leap Motion mit Vive/Hololens

##### 6.1.2 Unity Projekt

#### 6.2 Interaktion Hololens

- Prototyp für hololens Demo - Deckt Hologram Interaktionen ab
- Aufbau Szene - Funktionsweise der GameObjekte/ Skripte

##### 6.2.1 HoloToolkit

#### 6.3 Interaktion Vive

- Aufbau Szene - Funktionsweise der GameObjekte/ Skripte

##### 6.3.1 Leap Motion

#### 6.4 Volume Rendering

- Grundlage: GitHub Projekt

Wie in Kapitel 5 beschrieben, wurde der 3D-Darstellung des Gehirns mit Hilfe von Volume Rendering umgesetzt.

Im Kapitel ?? wurde der theoretische Vorgang dieser Technik beschrieben. Dieses Kapitel fokussiert sich auf die Implementierung der einzelnen Schritte.

Die volumetrische Darstellung der Gehirns wird im Grunde genommen mit nur 3 Skripten erzeugt. Zuerst wird in VolumeRendering.cs das Mesh des Würfeln generiert. Hier werden auch die Parameter aktualisiert, die für das Rendering relevant sind, wie z.B. die 3D-Textur oder Farbe, sowie die Parameter, die durch die Nutzerschnittstelle manipuliert werden können. Die Parameter werden an den Shader übergeben, in dem das Rendering definiert ist. Der Cg/HLSL Code, der den Vertex- und Fragmentshader implementiert ist dabei in ein eigenes Script ausgelagert.

##### 6.4.1 Shader in Unity

##### 6.4.2 Ray Casting

Im Fragmentshader wird zunächst ein Strahl definiert, der von dem aktuell betrachteten Vertex aus von der Kameraposition in die Welt "geschossen" wird. In einem selbst definierten struct

werden die maximalen und minimalen Werte definiert, aus denen sich die Eckpunkte des dargestellten Würfels zusammensetzen. Anschließend wird geprüft, ob der Strahl den Würfel schneidet.

Um die Schnittpunkte des Strahls zu ermitteln nimmt man an, dass die sechs Seiten des Würfels auf jeweils sechs Ebenen liegen, wobei davon zwei immer parallel sind. Zuerst werden jetzt alle Schnittpunkte des Strahls mit diesen Ebenen berechnet und dann geprüft, ob die Schnittpunkte innerhalb des Würfels liegen. Der Würfel wird durch zwei Eckpunkte beschrieben. Da der Würfel Koordinaten von -0,5 bis 0,5 hat können wir hierfür die jeweils kleinsten und größten Koordinaten nutzen. Die Schnittpunkte des Strahls, mit den x-, y- und z-Ebenen ergeben sich durch das Umstellen der Formel, die einen Strahl beschreibt:

$$p = r_{\text{Ursprung}} + t * r_{\text{Richtung}}$$

$r_{\text{Ursprung}}$  ist dabei der Ursprung des Strahls und  $r_{\text{Richtung}}$  seine Richtung.  $p$  ist ein Punkt auf dem Strahl und  $t$  ein Parameter, der bestimmt wie weit der Punkt vom Ursprung entfernt ist.

Um die Schnittpunkte mit den Ebenen zu erhalten wird die Formel nach  $t$  umgestellt:

$$t = (p - r_{\text{Ursprung}}) / r_{\text{Richtung}}$$

Für  $p$  werden jeweils die beiden Eckpunkte des Würfels eingesetzt, die den Würfel beschreiben. Dadurch sind in den zwei dreidimensionalen Vektoren  $t_{\text{unten}}$  und  $t_{\text{oben}}$  insgesamt 6 Schnittpunkte mit den Ebenen bekannt. Zwei auf jeder Ebene. Durch den Vergleich der t-Vektoren wird festgelegt, welcher Eckpunkt (und damit welche Ebene) weiter vorne liegt. Jetzt muss bestimmt werden, ob diese Schnittpunkte sich innerhalb des Würfels befinden. Dazu werden jeweils die x-, y- und z-Werte der t-Vektoren untereinander verglichen. Für den näher gelegenen t-Vektor wird der maximale bestimmt, für den weiter entfernten der minimale Wert bestimmt. Ist der Wert des näheren ts größer als der des entfernten, liegt der Schnittpunkt nicht in dem Würfel. Andersherum tut er es.

Die beiden t-Werte werden als  $t_{\text{nah}}$  und  $t_{\text{fern}}$  gespeichert. Mit dem Ursprung des Strahls und  $t_{\text{fern}}$  werden Anfang, Ende und die Länge des Strahls berechnet. Mit Hilfe der Länge kann ermittelt werden um wie weit pro Iteration am Strahl entlang gegangen werden soll. Dadurch wird der Strahl nur bis zu seinem Austritt aus dem Würfel abgetastet.

In einer for-Schleife wird jeder Strahl nun abgetastet. In jeder Iteration wird jeweils ein Punkt betrachtet. Der Punkt verschiebt sich entlang des Strahls um die zuvor berechnete Distanz. Für jeden Punkt werden zuerst die uv-Koordinaten berechnet, da sonst nur ein viertel des MRT-Bildes dargestellt würde. Für die Koordinaten werden dann die jeweiligen Isowerte aus der 3D-Textur gelesen, die zuvor mit den MRT-Bildern befüllt wurde.

Hierbei werden lediglich die uv-Koordinaten als Indices für die Textur verwendet. Der Isowert ist dabei im Alphakanal der Textur gespeichert. Da es sich nicht um eine Farbe sondern nur einen Grauwert handelt, können die anderen Farbkanäle der Textur mit der ?Magnitude? des jeweiligen Pixels befüllt werden. Darauf wird in der Sektion Transferfunktion genauer eingegangen. Der Isowert wird außerdem noch mit der Intensität multipliziert, die der Nutzer beeinflussen kann. An dieser Stelle wird aber auch geprüft, ob der betreffende Punkt überhaupt zu sehen ist, oder aufgrund der verschiebbaren Schichten nicht sichtbar sein sollte. Dazu wird zuerst der aktuell betrachtete Punkt mit der Rotationsmatrix des Modells multipliziert. Der Punkt wird dann mit den minimalen und maximalen x-, y-, und z-Werten der verschiebbaren Schicht verglichen. Das Ergebnis des Vergleichs wird dabei in einer Variable gespeichert. Ist der Punkt kleiner als das Minimum oder größer als das Maximum wird 0 gespeichert, ansonsten 1. Die beiden Werte werden anschließend mit dem Isowert multipliziert. Ist einer der Werte null, ist auch der ermittelte Isowert null, was im Alphakanal totale Transparenz bedeutet.

Dies tritt ein, da der Isowert zunächst für alle Farbkanäle verwendet wird, um eine neue Farbe zu deklarieren. ?

An dieser Stelle wird über die Transferfunktion der entsprechende Farbwert aus der zugehörigen Textur gelesen. Dazu wird der Isowert als Index verwendet. Die Funktionsweise und Implementierung der Transferfunktion wird in der Sektion 6.4.3 beschrieben. Die Transferfunktion wird nur abgerufen, wenn der Isowert nicht 0 ist, da sonst die Transparenz überschrieben würde. Ist die Farbe bekannt, wird der betrachtete Voxel illuminiert. Dies ist in der Sektion ?? beschrieben.

Der Alphawert der so erhaltenen Farbe wird noch einmal halbiert, um die Darstellung semi-transparent erscheinen zu lassen.

Schließlich wird der erhaltene Farbwert mit den vorhergehenden verrechnet. Die Komposition erfolgt dabei von vorne nach hinten, da der Strahl in dieser Richtung abgetastet wird, wie folgt:

$$\hat{C}_i = (1 - \hat{A}_{i-1})C_i + \hat{C}_{i-1}$$

$$\hat{A}_i = (1 - \hat{A}_{i-1})A_i + \hat{A}_{i-1}$$

Wobei  $\hat{C}_i$  die Farbe und  $\hat{A}_i$  die Transparenz der Farbe des vordersten Voxels ist. Wenn diese Farbe einen vorher definierten Schwellenwert überschreitet wird die Schleife abgebrochen. Der Schwellenwert kann vom Nutzer manipuliert werden und bestimmt die ?Helligkeit? der Darstellung.

Die Farbe wird schlussendlich noch auf einen Wert zwischen 0 und 1 festgesetzt und mit der Farbe der Maske verrechnet.

- Transferfunktion: Erstellen einer Textur
- Illumination: Gradient vector = normal - Gradient:

### 6.4.3 Transferfunktion

- lookup
- erzeugen von transferfunktion

### 6.4.4 Illumination

Das Volumen wird mit einem einfachen Beleuchtungsmodell illuminiert, wodurch es mehr Plastizität erhält. Gängiger Weise wird dafür das Phong Beleuchtungsmodell verwendet. Das Modell setzt sich aus drei Komponenten zusammen: Der ambienten und der diffusen Beleuchtung, sowie der spiegelnden Reflexion. In Abbildung ?? ist zu sehen, wie die einzelnen Komponenten isoliert aussehen und wie sie zusammengesetzt das Phong Beleuchtungsmodell ergeben. Dieses wird durch folgende Formel beschreiben:

$$I = k_a + I_L k_d (\vec{l} \cdot \vec{n}) + I_L k_s (\vec{r} \cdot \vec{v})^n$$

$I$  ist dabei die Farbe des beleuchteten Punktes.  $\vec{n}$  ist die Normale und  $\vec{l}$  ist der Richtungsvektor zum Licht.  $I_L$  ist die Intensität des Lichtes.  $\vec{r}$  und  $\vec{v}$  sind der Reflexionsvektor des einfallenden Lichtes und der Richtungsvektor zur Kamera. Weiterhin setzt sich die Formel aus dem ambienten, diffusen und spiegelndem Koeffizienten  $k_a$ ,  $k_d$  und  $k_s$  zusammen. Hinzu kommt der Exponent  $n$ , der konstant ist und die Stärke der Reflexion bestimmt. (?) Die oben beschriebenen Komponenten werden aufaddiert und sind somit durch jeweils ein +-Zeichen voneinander getrennt, wodurch sie gut voneinander zu trennen sind.

Die Koeffizienten werden im Shader durch Farben repräsentiert. Der diffuse Koeffizient ist dabei der vorher aus der Transferfunktion gelesene Farbwert. Um diesen nicht durch die ambiente



Beleuchtung zu verfälschen, wird er mit einem konstanten Faktor multipliziert, um eine abgedunkelte Farbe zu erhalten, die als ambierter Wert verwendet wird. Die Reflexion wird weiß dargestellt. Als Reflexionsexponent hat sich der Wert 10 als am besten erwiesen.

Um die Reflexion berechnen zu können müssen außerdem der Lichtvektor und die normale bekannt sein. Unterscheidung direktionales und Punktlicht?

#### **6.4.5 Gradientenberechnung**

## Kapitel 7

### Evaluation

#### 7.1 Vergleich mit Anforderungen

#### 7.2 Nutzertest

#### 7.3 Ergebnisse

## Kapitel 8

### Zusammenfassung und Fazit

#### 8.1 Zusammenfassung

#### 8.2 Ausblick

#### 8.3 Fazit