



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

28. Januar 2019

Zusammenfassung

...

Inhaltsverzeichnis

1	Einleitung	4
1.1	Motivation	4
1.2	Zielsetzung	5
1.3	Struktur dieser Arbeit	5
2	Verwandte Arbeiten und Technologien	7
2.1	AR in der Medizin	7
2.2	Visualisierung von MRT-Daten in AR/VR	7
2.3	MRT Volume Rendering	7
2.4	Interaktion in AR	7
3	Aktueller Stand der Technik	8
3.1	MRT	8
3.1.1	Verfahren	8
3.1.2	Datenverarbeitung	8
3.1.3	Abgrenzung CT	9
3.1.4	Datenformate	9
3.2	Dreidimensionale Darstellung von MRT-Daten	9
3.2.1	Volume Rendering	9
3.2.2	Shearwarp	9
3.2.3	Marching Cubes	9
3.2.4	Voxel	9
3.2.5	Andere	9
3.3	AR und VR	9
3.4	Verarbeitung von MRT-Daten	9
3.4.1	Arbeit eines Radiologen	9
3.4.2	Software in der Radiologie	10
3.5	Interaktion in AR/VR	10
3.5.1	Integrierte Nutzereingaben	10
3.5.2	Leap Motion	10
3.5.3	Andere	10
4	Anforderungsanalyse	11
4.1	Interviews	11
4.1.1	Anforderungen aus Interviews	11
4.2	User Stories	12
4.3	Technische Anforderungen	14
5	Konzept	15
5.1	3D Darstellung	15
5.2	Darstellung des gekennzeichneten Bereichs	15
5.3	Endgerät	15

5.4	Interaktion/ UI	16
5.4.1	Interaktion AR	16
5.4.2	Interaktion VR	16
5.5	Unterstützung von Dateiformaten	16
6	Implementierung	17
6.1	Aufbau Struktur des Projektes	17
6.1.1	Kombination von Leap Motion mit Vive/Hololens	17
6.1.2	Unity Projekt	17
6.2	Interaktion Hololens	17
6.2.1	HoloToolkit	17
6.3	Interaktion Vive	17
6.3.1	Leap Motion	17
6.4	Volume Rendering	17
6.4.1	Ray Casting	17
6.4.2	Transferfunktion	19
6.4.3	Illumination	19
6.4.4	Gradientenberechnung	19
7	Evaluation	20
7.1	Vergleich mit Anforderungen	20
7.2	Nutzertest	20
7.3	Ergebnisse	20
8	Zusammenfassung und Fazit	21
8.1	Zusammenfassung	21
8.2	Ausblick	21
8.3	Fazit	21

Abbildungsverzeichnis

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 Neurologen 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 Neurologen entspricht, werden Interviews mit einem Neurologen 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 Neurologen 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° 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 Dreidimensionale Darstellung von MRT-Daten

3.2.1 Volume Rendering

! Nur Grundlagen, nicht Implementierung

3D Texturen

Raymarching

Transferfunktionen

3.2.2 Shearwarp

3.2.3 Marching Cubes

3.2.4 Voxel

3.2.5 Andere

3.3 AR und VR

3.4 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.

3.4.1 Arbeit eines Radiologen

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

Schlaganfälle auf einem MRT

3.4.2 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

3.5 Interaktion in AR/VR

3.5.1 Integrierte Nutzereingaben

3.5.2 Leap Motion

3.5.3 Andere

Kapitel 4

Anforderungsanalyse

4.1 Interviews

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

Im Rahmen des ersten Interviews wurde zunächst der Nutzen der Anwendung beschreiben. Diese soll Neurologen im Bereich der Schlaganfallprävention unterstützen. Wie im Kapitel Grundlagen beschrieben untersucht der Neurologe dabei die Abbildungen des Gehirns eines Patienten, die durch den MRT-Scan erzeugt wurden. Auffällige Bereiche, die auf einen Schlaganfall hindeuten könnten werden dabei markiert. Diese Darstellung soll nun inklusive des markierten Bereichs dreidimensional dargestellt werden. Durch die zusätzliche Dimension soll sowohl für Ärzte als auch Patienten die Ausmaße des betroffenen Bereich verdeutlichen, was die Einschätzung eines Schlaganfallpatienten verbessern könnte. Weiterhin könnte man anhand des Modells auch Voraussagen von Therapieerfolgen anschaulich demonstrieren.

Die Anwendung soll dabei als Prototyp dienen, die die Möglichkeiten, sowie das Potential für die Verwendung in der Praxis testet. Im Vordergrund steht hierbei die Darstellung in 3D, inklusive des markierten Bereichs.

Aus dem ersten Interview ließen sich außerdem folgende Anforderungen ableiten:

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	hoch
A03	3D-Darstellung des Gehirns, die auch Schlaganfallsbereich in 3D Erkennbar werden lässt	hoch
A04	Scrollen durch Schichten der 3D-Darstellung auf mindestens einer Achse	mittel
A05	Erkennbarkeit der Struktur inneren Struktur des Gehirns (entsprechend der MRT-Bilder)	hoch
A06	Scrollbare 2D Darstellung der MRT-Bilder auch mindestens einer Achse	mittel
A07	Manipulation der Darstellung ähnlich wie bei einem Hologramm.	niedrig
A10	Interaktionselemente sollten die Darstellung nicht verdecken	niedrig
A11	Scrollen durch Verwendung eines Scrollrads	niedrig
A12	Auswahl verschiedener MRT-Sequenzen (falls vorhanden)	niedrig
A13	Unterstützung von nifti-Daten	mittel
A14	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 fliehet (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 geupdated, 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 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 Würfeln 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_{Ursprung} ist dabei der Ursprung des Strahls und r_{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_{unten} und t_{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_{nah} und t_{fern} gespeichert. Mit dem Ursprung des Strahls und t_{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 6.4.2 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:

6.4.2 Transferfunktion

- lookup
- erzeugen von transferfunktion

6.4.3 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 ambienter 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?

6.4.4 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