



UNIVERSITÄT  
HEIDELBERG  
ZUKUNFT  
SEIT 1386

**HHTN**  
HOCHSCHULE HEILBRONN  
TECHNIK | WIRTSCHAFT | INFORMATIK

# Virtual-Reality in der Strahlentherapieaufklärung

Erste Untersuchungen zu Chancen und  
Herausforderungen der Digitalisierungs- und  
Simulationstechnologien

Master-Thesis

zur Erlangung des akademischen Grades  
Master of Science (M.Sc.) im Studiengang  
Medizinische Informatik

vorgelegt von

Julian Suleder  
192809

2. November 2017

**Referent:** Prof. Dr. Rolf Bendl, Hochschule Heilbronn

**Korreferent:** Dr. Kristina Giske, Deutsches Krebsforschungszentrum Heidelberg

**dkfz.**

DEUTSCHES  
KREBSFORSCHUNGZENTRUM  
IN DER HELMHOLTZ-GEMEINSCHAFT



Forschen für ein Leben ohne Krebs



# Zusammenfassung

**Titel:** Virtual-Reality in der Strahlentherapieaufklärung: Erste Untersuchungen zu Chancen und Herausforderungen der Digitalisierungs- und Simulationstechnologien

**Autor:** Julian Suleder (192809)

**Studium:** Medizinische Informatik

**Abschluss:** Master of Science (M.Sc.)

**Universität:** Hochschule Heilbronn & Universität Heidelberg

**Datum:** 2. November 2017

Im Mittelpunkt dieser Arbeit steht die Entwicklung eines Verfahrens für das Erzeugen von möglichst realitätsnahen Telepräsenzsimulationen für die Information von Patienten in der Strahlentherapie sowie die Anwendung des Verfahrens in der Entwicklung einer VR-Applikation auf Basis eines Demonstrators.

Nach einer Literaturrecherche bezüglich des aktuellen Stands der Aufklärung von Patienten wurden die Grundlagen von Virtual und Augmented Reality hinsichtlich der gegebenen Hardware ermittelt. Hierauf folgte die Auswahl von Software für das Scannen von realen Objekten mit dem gegebenen Tablet in einem Bestrahlungsraum des DKFZ sowie die Auswahl einer Game Engine für die Entwicklung des Demonstrators. Daraufhin wurde ein Rekonstruktionsalgorithmus ausgewählt. Anschließend wurden verschiedene Objekte im Bestrahlungsraum gescannt, sodass die Parameter des Algorithmus iterativ hinsichtlich der Qualität der erzeugten Objekte für den Einsatz in einer VR-Anwendung optimiert werden konnten. Daraufhin erfolgte eine Texturierung der Oberfläche mit Kamerafotos. Nach einer Aufbereitung der Modelle wurden diese in ein Virtual Environment importiert. Parallel dazu wurde nach der Auswahl der Unreal Engine als Game Engine, der Demonstrator entwickelt, in welchen die gescannten Modelle integriert wurden.

Das Verfahren liefert ausreichend genaue Ergebnisse, um Konzepte in der Strahlentherapieaufklärung vermitteln zu können. Der Effekt und die Akzeptanz der Technik spielen eine weitere wichtige Rolle für den Einsatz der Methodik und müssen durch eine Evaluation im klinischen Alltag validiert werden, wofür die Entwicklung einer klinisch anwendbaren Software auf Basis der gewonnenen Erkenntnisse notwendig wird.

# Abstract

**Title:** Virtual Reality in Radiation Therapy Patient Information: First Investigations on the Opportunities and Challenges of Digitalization and Simulation Technologies

**Author:** Julian Suleder (192809)

**Subject:** Medical Informatics

**Degree:** Master of Science (M.Sc.)

**University:** Heilbronn University & Heidelberg University

**Date:** November 2, 2017

The main objective of this work is the development of a method for the generation of realistic telepresence simulations for the information of patients in radiotherapy as well as the evaluation of this method within the development of a VR application as a demonstrator.

After a literature review regarding the current state of patient education, the basics of virtual and augmented reality were determined in terms of the given hardware. This was followed by the selection of software for the scanning of real objects with the given tablet in an irradiation room of the DKFZ and the selection of a game engine for the development of the demonstrator. Then a reconstruction algorithm was selected. Subsequently, various objects in the irradiation room were scanned so that the parameters of the algorithm could be iteratively optimized for the quality of the created objects for use in a VR application. After that, the surfaces were texturized with camera photos. After preparing the models, they were imported into a virtual environment. At the same time, after the selection of the Unreal Engine as a game engine, the demonstrator, in which the scanned models were integrated, was developed.

The method provides sufficiently accurate results to convey concepts in radiation therapy. The effect and acceptance of the technique play an important role in the use of the method and need to be validated in the clinical daily routine. The development of a clinically applicable software on the basis of the obtained knowledge is necessary.

## Abkürzungsverzeichnis

|                 |   |
|-----------------|---|
| <b>API</b>      | Application Programming Interface                     |
| <b>AR</b>       | Augmented Reality                                     |
| <b>CAD</b>      | Computer-Aided Design                                 |
| <b>CNR</b>      | Nationaler Forschungsrat                              |
| <b>CPU</b>      | Central Processing Unit                               |
| <b>DKFZ</b>     | Deutsches Krebsforschungszentrum                      |
| <b>fMRT</b>     | funktionelle Magnetresonanztomographie                |
| <b>FPS</b>      | Frames per second                                     |
| <b>GP3</b>      | Greedy-Projection Triangulation                       |
| <b>HIT</b>      | Heidelberger Ionenstrahl-Therapiezentrum              |
| <b>HMD</b>      | Head-Mounted Display                                  |
| <b>ISTI</b>     | Institut für Informationswissenschaft und Technologie |
| <b>LINAC</b>    | Elektronenlinearbeschleuniger                         |
| <b>MLC</b>      | Multi-Leaf-Kollimator                                 |
| <b>MRT</b>      | Magnetresonanztomographie                             |
| <b>PET</b>      | Positronen-Emissions-Tomographie                      |
| <b>PSR</b>      | Poisson Surface Reconstruction                        |
| <b>RGB</b>      | Farbraum aus den Farben Rot, Grün und Blau            |
| <b>RTAB-Map</b> | Real-Time Appearance-Based Mapping                    |
| <b>SBA</b>      | Sparse-Bundle Adjustment                              |
| <b>SLAM</b>     | Simultaneous Localization And Mapping                 |
| <b>SPECT</b>    | Single Photon Emission Computed Tomography            |
| <b>SPSR</b>     | Screened Poisson Surface Reconstruction               |
| <b>ToF</b>      | Time of Flight  |
| <b>VB</b>       | Virtual Body  |
| <b>VE</b>       | Virtual Environment                                   |
| <b>VR</b>       | Virtual Reality                                       |

↗ Link zu einem 3D-Modell. Hinweis: Die Texturen der Modelle besitzen ggf. eine reduzierte Größe von 1024x1024 Pixel<sup>2</sup>, da die verwendeten Plattformen nur Texturen bis zu dieser Größe unterstützen. Das ist eine Reduzierung auf 1/64 der originalen Größe, sodass der visuelle Eindruck beeinträchtigt sein kann.

# Abbildungsverzeichnis

|      |   |    |
|------|---|----|
| 2.1  | Elektronenlinearbeschleuniger im DKFZ in Heidelberg . . . . .   | 5  |
| 2.2  | Strahlenführung am Heidelberger-Schwerionen-Therapiezentrum . . . . .   | 6  |
| 2.3  | Computertomograph im DKFZ und eine Aufnahme eines Hirntumors . . . . .  | 7  |
| 2.3a | Computertomograph im DKFZ in Heidelberg . . . . .   | 7  |
| 2.3b | CT-Aufnahme: Tumor in der rechten Hirnhälfte . . . . .  | 7  |
| 2.4  | Tumor im rechten Okzipitallappen . . . . .  | 8  |
| 2.4a | MRT-Aufnahme: Tumor in der rechten Hirnhälfte (T1) . . . . .  | 8  |
| 2.4b | MRT-Aufnahme: Tumor in der rechten Hirnhälfte (T2) . . . . .  | 8  |
| 2.5  | Beispiel einer Augmented Reality Anwendung . . . . .  | 16 |
| 2.6  | 3D-Wireframe eines Würfels mit Vertices und Edges . . . . .   | 18 |
| 2.7  | Aufbau eines Szenengraphen mit Universum und einem Fahrrad . . . . .  | 19 |
| 3.1  | Modellieren eines 3D-Charakters in Blender . . . . .  | 22 |
| 3.2  | Loop Closure Detection und visuelles Feedback beim 3D-Scannen . . . . .   | 24 |
| 3.2a | Loop Closure Detection bei der Schätzung der Tablet-Posen . . . . .   | 24 |
| 3.2b | 3D-Scannen mit der RTAB-Map Android-App . . . . .   | 24 |
| 3.3  | Darstellung eines 3D-Modells mit Meshlab . . . . .  | 25 |
| 3.4  | Darstellung der Kamera-Kegelstümpfe für das Sparse-Bundle-Adjustment  | 26 |
| 3.5  | SPSR: Wasserdichte Oberflächenrekonstruktion durch Octrees . . . . .  | 27 |
| 3.5a | Ein Octree repräsentiert als Baum und partiell geteilter Würfel   | 27 |
| 3.5b | 3D-Rekonstruktion eines Drachen mit der SPSR . . . . .  | 27 |
| 3.6  | 3D-Rekonstruktion eines Arbeitsplatzes mit der GP3 . . . . .  | 28 |
| 3.7  | Lenovo Phab 2 Pro mit einer Augmented Reality App . . . . .   | 29 |
| 3.8  | Berechnung der euklidischen Distanz mit Blueprints . . . . .  | 32 |
| 4.1  | Der Workflow des 3D-Scannens . . . . .  | 34 |
| 4.2  | 3D-Scannen mit der RTAB-Map Android-App . . . . .   | 35 |
| 4.3  | Foto und kolorierte Punktwolke eines Computertomographen nach der Aufnahme mit der RTAB-Map Android-App . . . . . | 37 |
| 4.3a | Foto des realen Computertomographen . . . . .   | 37 |
| 4.3b | Kolorierte Punktwolke des gescannten Computertomographen . . . . .  | 37 |
| 4.4  | Ergebnisse der Rekonstruktion des Computertomographen . . . . .   | 39 |
| 4.4a | Greedy Projection Triangulation des Computertomographen . . . . .   | 39 |
| 4.4b | SPSR des Computertomographen mit minimaler Octree-Tiefe 6   | 39 |
| 4.4c | SPSR des Computertomographen mit minimaler Octree-Tiefe 9   | 39 |
| 4.5  | Textur zusammengesetzt aus Kamerabildern . . . . .  | 40 |
| 4.6  | Ausschnitt einer Textur in einem VE . . . . .   | 41 |
| 4.7  | Modell mit Faces ohne Textur . . . . .  | 42 |
| 4.8  | Manuelles Veredeln des gescannten Datensatzes mit Meshlab . . . . .   | 43 |
| 4.8a | Rekonstruktion mit Artefakten . . . . .   | 43 |
| 4.8b | Manuell gesäuberte Rekonstruktion . . . . .   | 43 |

## Abbildungsverzeichnis

---

|      |   |    |
|------|---|----|
| 4.9  | Minimaler Aufbau der VR-Hardware für eine Demonstration . . . . .       | 44 |
| 4.10 | Vogelperspektive der Szene im Demonstrator . . . . .                    | 47 |
| 4.11 | Aufbau der funktionalen Aktoren der 3D-Szene . . . . .                  | 49 |
| 4.12 | Datenmodell: UML Klassendiagramm der Anwendungsdomain . . . . .         | 50 |
| 4.13 | Blueprint: Hinzufügen einer Konfiguration . . . . .                     | 52 |
| 4.14 | Blueprint: Konfiguration für die Bewegung der Gantry . . . . .          | 53 |
| 4.15 | Blueprint: Konfiguration für die Bewegung der Patientenliege . . . . .  | 54 |
| 4.16 | Blueprint: Konfiguration für die Bestrahlung . . . . .                  | 55 |
| 4.17 | Blueprint: Konfigurationsfolge für eine Bestrahlung . . . . .           | 56 |
| 4.18 | Timeline der Bewegung der Gantry . . . . .                              | 58 |
| 4.19 | Event für das Bewegen der Gantry . . . . .                              | 58 |
| 4.20 | Event für das Bewegen des horizontalen Segments der Patientenliege . .  | 59 |
| 4.21 | Aktualisierung des Zählers für das Warten von Simulationsmodellen .     | 60 |
| 4.22 | Tick-Event des Linearbeschleunigers als Zustandsautomat . . . . .       | 61 |
| 4.23 | Blueprint für das Feststellen des aktuellen Zustands des Beschleunigers | 62 |
| 4.24 | Material des pulsierenden Bestrahlungsstrahls aus zwei Texturen . . . . | 63 |
| 4.25 | Partikelsystem des Bestrahlungsstrahl im Demonstrator . . . . .         | 64 |
| 4.26 | Approximation einer Objektoberfläche bei Strahlkollision . . . . .      | 64 |
| 4.27 | Blueprint der Spurverfolgung bei Laser und Bestrahlungsstrahl . . . . . | 65 |
| 4.28 | Stoppen eines Ambient Sound Aktors mit einem Blueprint . . . . .        | 66 |

# Inhaltsverzeichnis

|  |           |
|--|-----------|
| <b>Abkürzungsverzeichnis</b>   | <b>v</b>  |
| <b>Abbildungsverzeichnis</b>   | <b>vi</b> |
| <b>1 Einführung</b>  | <b>1</b>  |
| 1.1 Motivation . . . . .   | 1         |
| 1.2 Ziel der Arbeit . . . . .  | 1         |
| 1.3 Vorgehensweise und Struktur der Arbeit . . . . .                           | 2         |
| <b>2 Grundlagen</b>  | <b>3</b>  |
| 2.1 Strahlentherapie . . . . .   | 3         |
| 2.1.1 Wirkungsprinzip und Zielsetzung . . . . .                                | 3         |
| 2.1.2 Indikation und Kombination mit anderen Therapieformen . . . . .          | 4         |
| 2.1.3 Gerätetechnik in der Teletherapie . . . . .                              | 4         |
| 2.1.3.1 Elektronenlinearbeschleuniger . . . . .                                | 4         |
| 2.1.3.2 Bestrahlung mit Ionen . . . . .  | 5         |
| 2.1.4 Bildgebung in der Strahlentherapie . . . . .                             | 6         |
| 2.1.4.1 Computertomographie . . . . .  | 7         |
| 2.1.4.2 MR-Bildgebung . . . . .  | 8         |
| 2.2 Aufklärung und Information von Patienten in der Strahlentherapie . . . . . | 9         |
| 2.2.1 Prozess der Therapie und Information des Patienten . . . . .             | 9         |
| 2.2.2 Informationsbedarf der Patienten . . . . .                               | 9         |
| 2.2.3 Angst in der Therapie . . . . .  | 10        |
| 2.2.4 Einsatz und Effekt von Audioaufnahmen . . . . .                          | 11        |
| 2.2.5 Einsatz und Effekt von Videos und Simulationen . . . . .                 | 12        |
| 2.3 Virtuelle Realität . . . . .   | 13        |
| 2.3.1 Virtuelle Umgebung . . . . .   | 13        |
| 2.3.2 Präsenz . . . . .  | 14        |
| 2.3.2.1 Ortsillusion und Virtual Body . . . . .                                | 14        |
| 2.3.2.2 Plausibilitätsillusion . . . . .                                       | 14        |
| 2.3.3 Immersion . . . . .  | 14        |
| 2.3.4 Willentliches Ausblenden des Unglaubens . . . . .                        | 15        |
| 2.4 Augmented Reality (AR) . . . . .   | 15        |
| 2.5 Einsatz von VR & AR in der Medizin . . . . .                               | 16        |
| 2.6 Spieleentwicklung mit Game Engines . . . . .                               | 17        |
| 2.6.1 3D-Objekte aus Vertices, Edges, Faces und Normalen . . . . .             | 17        |
| 2.6.2 Aufbau einer 3D-Szene . . . . .  | 18        |
| 2.6.3 Inhalte einer 3D-Szene . . . . .   | 19        |
| 2.6.4 Game Engines . . . . .   | 20        |

|  |           |
|--|-----------|
| <b>3 Material und Methoden</b>                                     | <b>21</b> |
| 3.1 Verfahren zur Digitalisierung realer Objekte . . . . .         | 21        |
| 3.1.1 3D-Scannen . . . . .   | 21        |
| 3.1.2 Modellierung von 3D-Objekten . . . . .                       | 22        |
| 3.2 Software der Digitalisierung realer Objekte . . . . .          | 23        |
| 3.2.1 Blender . . . . .  | 23        |
| 3.2.2 RTAB-Map . . . . .   | 23        |
| 3.2.3 MeshLab . . . . .  | 25        |
| 3.3 Algorithmen . . . . .  | 26        |
| 3.3.1 Sparse-Bundle Adjustment (SBA) . . . . .                     | 26        |
| 3.3.2 Screened Poisson Surface Rekonstruktion (SPSR) . . . . .     | 27        |
| 3.3.3 Fast Greedy Projection Triangulation (GP3) . . . . .         | 28        |
| 3.4 AR mit Google's Project Tango & Lenovo's Phab 2 Pro . . . . .  | 29        |
| 3.5 Software zur Entwicklung von VR-Anwendungen . . . . .          | 30        |
| 3.5.1 Unity 3D . . . . .   | 30        |
| 3.5.2 Unreal Engine 4 . . . . .                                    | 30        |
| 3.5.2.1 Aufbau eines Levels . . . . .                              | 31        |
| 3.5.2.2 Simulation und Events . . . . .                            | 31        |
| 3.5.2.3 Die Blueprint-Sprache . . . . .                            | 32        |
| 3.5.2.4 Erstellen von Aktoren . . . . .                            | 32        |
| <b>4 Ergebnisse</b>  | <b>33</b> |
| 4.1 Konzept der Information und Aufklärung von Patienten . . . . . | 33        |
| 4.2 3D-Scannen von Objekten . . . . .                              | 34        |
| 4.2.1 Anforderungen und Auswahl . . . . .                          | 34        |
| 4.2.2 Der Scanvorgang . . . . .                                    | 35        |
| 4.2.3 Sparse-Bundle Adjustment . . . . .                           | 36        |
| 4.2.4 Rekonstruktion . . . . .                                     | 36        |
| 4.2.4.1 Das Vergleichsobjekt . . . . .                             | 36        |
| 4.2.4.2 Parametrisierung der Rekonstruktionen . . . . .            | 37        |
| 4.2.4.3 Ergebnisse der Rekonstruktionen . . . . .                  | 38        |
| 4.2.5 Textur-Mapping . . . . .                                     | 40        |
| 4.2.6 Löschen von Faces ohne Textur . . . . .                      | 41        |
| 4.2.7 Reduzierung der Faces . . . . .                              | 42        |
| 4.2.8 Veredelung des 3D-Modells . . . . .                          | 43        |
| 4.3 Aufbau eines HTC Vive VR-Systems . . . . .                     | 44        |
| 4.4 Auswahl einer Game Engine . . . . .                            | 45        |
| 4.5 Entwicklung des Demonstrators . . . . .                        | 46        |
| 4.5.1 Funktionalität des Demonstrators . . . . .                   | 46        |
| 4.5.2 Aktoren der Szene . . . . .                                  | 48        |
| 4.5.3 Datenmodell für Animation und Interaktion . . . . .          | 50        |
| 4.5.4 Parametrisierung von Bewegungsabläufen . . . . .             | 51        |

|                  |   |           |
|------------------|---|-----------|
| 4.5.5            | Drehung der Gantry . . . . .  | 56        |
| 4.5.6            | Bewegen der Patientenliege . . . . .                                | 59        |
| 4.5.7            | Aktives Warten von Aktoren . . . . .                                | 60        |
| 4.5.8            | Zustände des Linearbeschleunigers . . . . .                         | 60        |
| 4.5.9            | Bestrahlungsstrahl und Laser als Partikelsystem . . . . .           | 62        |
| 4.5.10           | Spurverfolgung bei Laser und Bestrahlungsstrahl . . . . .           | 65        |
| 4.5.11           | Objekt- und Hintergrundgeräusche . . . . .                          | 66        |
| <b>5</b>         | <b>Diskussion</b>   | <b>67</b> |
| 5.1              | Konzept zur Patientenaufklärung mit Virtual Reality . . . . .       | 67        |
| 5.2              | Verfahren zur Digitalisierung von Strahlentherapie-Szenen . . . . . | 68        |
| 5.2.1            | Das Vergleichsobjekt . . . . .                                      | 68        |
| 5.2.2            | Qualität der Scans mit dem Tablet . . . . .                         | 69        |
| 5.2.3            | Auswahl und Parametrisierung der Rekonstruktion . . . . .           | 70        |
| 5.2.4            | Bewertung und Probleme des Textur-Mapping . . . . .                 | 70        |
| 5.2.5            | Reduzierung der Faces . . . . .                                     | 71        |
| 5.3              | Erfahrungen aus dem Aufbau des HTC Vive VR-Systems . . . . .        | 71        |
| 5.4              | Entwicklung des Demonstrators mit der Unreal Engine . . . . .       | 72        |
| 5.4.1            | Datenmodell und Gruppierung von Aktoren . . . . .                   | 72        |
| 5.4.2            | Beleuchtung . . . . .   | 72        |
| 5.4.3            | Kollimation und Darstellung von Strahlenform . . . . .              | 72        |
| 5.4.4            | Darstellung der Positionierungslaser . . . . .                      | 73        |
| 5.5              | Ausblick . . . . .  | 73        |
| <b>Literatur</b> |   | <b>75</b> |

# **1 Einführung**

Im Mittelpunkt dieser Arbeit steht die Entwicklung eines Verfahrens für das Erzeugen von möglichst realitätsnahen Telepräsenzsimulationen für die Information von Patienten in der Strahlentherapie sowie die Anwendung des Verfahrens in der Entwicklung einer VR-Applikation auf Basis eines Demonstrators.

## **1.1 Motivation**

Der Prozess und die viele hochspezialisierte Technik der Strahlentherapie können herausfordernd und im Kontext der Erkrankung für den Patienten belastend sein. Die krankheitsbedingte Bedrohung, die zum Einsatz kommende komplexe Technik und die resultierenden Gefahren verursachen Ängste bei den Patienten. Um diese Ängste zu reduzieren und den Patienten über Risiken und Aussichten der Therapie zu informieren, ist eine ausführliche Aufklärung erforderlich und vorgeschrieben. Bei der Therapie selbst bleiben meist weniger als fünf Minuten, um den Patienten auf die nachfolgende Diagnostik oder Therapie vorzubereiten. Eine ausgiebige Aufklärung der Funktionsweise, Rechtfertigung der Geräte und Maßnahmen erfolgt meist kurz vor oder während der Therapiesitzung durch die betreuenden Medizinisch-technischen (Radiologie-)Assistenten. Dabei wird meist nur auf für die aktuelle Sitzung relevante Dinge eingegangen. Durch aktuell aufstrebende Technik, wie z.B. Virtual-Reality-Brillen, kann es möglich sein, eine Aufklärung der Patienten digital durchzuführen und eventuelle Vorbehalte und Ängste zu reduzieren.

## **1.2 Ziel der Arbeit**

Die Ziele dieser Masterarbeit sind in der nachfolgenden Aufzählung erläutert:

1. Konzeption eines Verfahrens für die Digitalisierung von realen Objekten in der Strahlentherapie unter Berücksichtigung des Anwendungsgebiets und bereits vorhandener Virtual und Augmented Reality Hardware
2. Evaluation des Verfahrens in einem Bestrahlungsraum des Deutschen Krebsforschungszentrums in Heidelberg
3. Das Identifizieren von Faktoren, welche die Aufklärung und Information von Patienten in der Strahlentherapie beeinflussen und aufzeigen, inwieweit diese durch den Einsatz einer immersiven VR-Anwendung positiv ausgenutzt werden können

4. Implementierung eines Demonstrators, der nach 1. entwickelte Verfahren in Anwendung zeigt und motiviert, wie die in 3. identifizierten Faktoren umgesetzt werden können.
5. Ziel ist es nicht, eine Anwendung zu entwickeln, die direkt im klinischen Umfeld Anwendung finden kann oder einen Inhalt aus der Strahlentherapie erklärt, sondern herauszufinden, welche Vor- und Nachteile, Chancen und Hürden ein solches System bzw. Verfahren hinsichtlich der technischen Umsetzung besitzt

### 1.3 Vorgehensweise und Struktur der Arbeit

Zu Beginn der Arbeit wurde der aktuelle Stand der Aufklärung von Patienten durch eine Literaturrecherche identifiziert und die Grundlagen von Virtual und Augmented Reality hinsichtlich der gegebenen Hardware ermittelt. Hierauf folgte die Auswahl von Software für das Scannen von realen Objekten mit dem gegebenen Tablet in einem Bestrahlungsraum des Deutschen Krebsforschungszentrums (DKFZ) sowie die Auswahl einer Game Engine für die Entwicklung des Demonstrators. Daraufhin wurde nach einer Analyse der Leistungsfähigkeit von Software und Hardware ein Rekonstruktionsalgorithmus ausgewählt. Anschließend wurden verschiedene Objekte im Bestrahlungsraum gescannt, sodass die Parameter des Algorithmus iterativ hinsichtlich der Qualität der erzeugten Objekte für den Einsatz in einer VR-Anwendung optimiert werden konnten. Parallel dazu wurde nach der Auswahl der Game Engine der Demonstrator, in welchen nach und nach die gescannten Modelle integriert wurden, entwickelt.

Diese Arbeit gliedert sich nachfolgend in die Literaturrecherche zu den Erkenntnissen und Methoden der Patientenaufklärung, Grundlagen des Anwendungsgebiets der Strahlentherapie sowie den Grundlagen von Virtual und Augmented Reality hinsichtlich der Ziele dieser Arbeit (Kapitel 2). In Kapitel 3 werden die verwendeten Konzepte, Software, Algorithmen und Hardware vorgestellt. Im Anschluss daran finden sich in Kapitel 4 das Digitalisierungsverfahren, unter Darlegung der Auswahl des Algorithmus zur Rekonstruktion der Scan-Daten, dessen Parametrisierung und weitere Schritte, die notwendig sind, um die Modelle für die VR-Anwendung nutzbar zu machen. Weiterhin wird in Kapitel 4 das Konzept der Patientenaufklärung mit Virtual Reality, welches auf den Erkenntnissen der Literaturrecherche aufbaut, vorgestellt und die Realisierung des Demonstrators dargelegt. Kapitel 5 bildet mit der Diskussion der Ergebnisse und dem Ausblick den Abschluss dieser Arbeit.

## 2 Grundlagen

In diesem Kapitel werden zunächst die Grundlagen der Strahlentherapie (2.1) vorgestellt, um ein Verständnis der Anwendungsdomäne der in dieser Arbeit konzipierten Anwendung zu schaffen und hinsichtlich der in Kapitel 1 dargelegten Problemstellung zu sensibilisieren. Anschließend wird auf die Grundlagen der Virtuellen Realität (2.3) und die Entwicklung von Spielen mit Game Engines (2.6) eingegangen, um die Konzepte der neuen Technologie im Hinblick auf das entwickelte Verfahren und die Realisierung im Demonstrator zu erläutern.

### 2.1 Strahlentherapie

Im nachfolgenden Unterabschnitt (2.1.1) wird zunächst das Wirkungsprinzip und die Zielsetzung der Strahlentherapie sowie in Unterabschnitt 2.1.2 deren Einsatz in der Krebstherapie erläutert. Anschließend wird auf die Gerätetechnik (2.1.3) und Bildgebung (2.1.4) in der Therapie eingegangen.

#### 2.1.1 Wirkungsprinzip und Zielsetzung

Das Wirkungsprinzip der Strahlentherapie ist nach Wannenmacher et al. die Energieübertragung an biologische Gewebe durch Partikelstrahlung (Elektronen, Protonen, Neutronen, Ionen) oder elektromagnetische Photonenstrahlung. Das physikalische Maß für diese Energieabgabe ist die Energiedosis in der Einheit Gray (1 Gy), die als der Quotient aus absorbierte Strahlungsenergie und Masse des Materials definiert wird. Der positive Therapieeffekt durch Erhöhung der Dosis sinkt mit zunehmender Dosis, da auch das Risiko der Schädigung von gesundem Gewebe zunimmt. Um dennoch genug Dosis im Gewebe zu platzieren, wird die Dosis fraktioniert, d.h. in mehreren Einzeldosen, verabreicht. Die reine Bestrahlungszeit einer Fraktion liegt meist zwischen zwei und zehn Minuten. Das gesunde Gewebe kann sich im Gegensatz zum Tumorgewebe, das zunehmend abgetötet wird, in der Zeit zwischen den Anwendungen durch Reparaturmechanismen erholen. Das Ziel der Strahlentherapie ist es, das Tumorgewebe durch eine Narbe des Gefäßbindegewebes zu ersetzen, nicht aber jedoch ein pathologisches Absterben der Gewebe mit entsprechenden Komplikationen zu erzeugen [Wan13, S. 4 ff.].

### 2.1.2 Indikation und Kombination mit anderen Therapieformen

Die Anwendung der Strahlentherapie ist abhängig von der Gewebestruktur, Lokalisierung und Ausbreitung der Erkrankung, der Möglichkeit von Operation und Chemotherapie sowie der Belastbarkeit des Patienten. Die alleinige Strahlentherapie wird bei strahlensensiblen, regional begrenzten Tumoren, ungünstig lokalisierten Tumoren oder inoperablen Hirntumoren angewandt. Eine Bestrahlung kann auch vor oder nach einem operativen Eingriff erfolgen, um den Tumor zu verkleinern, Organe und die Funktionsfähigkeit zu erhalten oder eine Verschleppung von Tumorzellen zu verhindern [Wan13, S. 7].

### 2.1.3 Gerätetechnik in der Teletherapie

Am häufigsten erfolgt die Strahlentherapie mittels außerhalb des Körpers gelegenen Strahlenquellen, der so genannten Teletherapie. In der Teletherapie werden verschiedenste Bestrahlungsgeräte, wie z.B. in der Vergangenheit Isotopenbestrahlungsgeräte mit den Isotopen  $^{137}\text{Cäsium}$  oder  $^{60}\text{Cobalt}$ , eingesetzt. Diese wurden gegen Anfang des neuen Jahrtausends durch Elektronenbeschleuniger (2.1.3.1) ersetzt. Eine andere Art der Strahlentherapie basiert auf der Bestrahlung von beschleunigten, geladenen Teilchen (2.1.3.2). Bei beiden Therapieformen ist eine exakte Positionierung des Patienten notwendig, damit punktgenau bestrahlt und das gesunde Gewebe geschont werden kann.

#### 2.1.3.1 Elektronenlinearbeschleuniger

Das Funktionsprinzip eines Elektronenlinearbeschleunigers beruht auf Elektronen, die in einem Magnetfeld beschleunigt werden und auf eine Wolfram-Scheibe prallen, wodurch elektromagnetische Strahlung emittiert wird [vgl. Wan13, S. 35 ff.]. Ein Elektronenlinearbeschleuniger (LINAC) besteht aus einem feststehenden Teil und einem beweglichen Bestrahlungsarm (Gantry), in welchem sich unter anderem das Beschleunigerrohr und der Strahlerkopf, welcher für das Ablenken des Strahls in Richtung des Isozentrums verantwortlich ist, befinden (siehe Abbildung 2.1). Der Strahl wird durch eine Begrenzung in seiner Form eingeschränkt. Dieser Multi-Leaf-Kollimator (MLC) besteht aus mehreren Paaren von Wolfram-Blenden, die unabhängig voneinander bewegbar sind. Dies ermöglicht es dynamisch die Form des Behandlungsstrahls an die des Zielvolumens anzupassen [vgl. Wan13, S. 41 ff.]. Das Isozentrum bildet den Ursprung aller Koordinatensysteme, in denen mit entsprechenden Offsets und Winkeln versehen, andere Bauteile, wie z.B. Gantry, Patientenliege und Kollimator, spezifiziert werden. Der Patient wird auf einer Liege so positioniert, dass sich das Zentrum des zu bestrahlenden Gebiets im Isozentrum des LINACs befindet. Dazu werden verschiedene patientenspezifische Lagerungshilfen angefertigt. Der Patient spürt den Strahl während der Therapie nicht, nur selten treten nach einer Sitzung Hautreizungen auf.



Abbildung 2.1: Ein Linearbeschleuniger, wie er für die Forschung im DKFZ in Heidelberg verwendet wird (rechts) sowie die gedrehte Patientenliege (links).

### 2.1.3.2 Bestrahlung mit Ionen

Zusätzlich zur Photonenbestrahlung gibt es die Protonen- und Schwerionentherapie. Bei der Protonentherapie wird mit geladenen Kernen von Wasserstoffatomen, bei der Schwerionentherapie mit Kohlenstoffionen bestrahlt [Med16]. Als Ionen werden geladene Teilchen bezeichnet. Für diese Bestrahlungsarten wird ein Teilchenbeschleuniger benötigt, da die Teilchen auf etwa 75% der Lichtgeschwindigkeit beschleunigt werden müssen. Beide Therapien können mit der gleichen Therapieanlage durchgeführt werden, da die Art der in die Anlage eingespeisten Ionen verändert werden kann [Med16]. Das Heidelberger Ionenstrahl-Therapiezentrum (HIT) ist beispielhaft eines von wenigen Zentren weltweit, das die Bestrahlung mit Schwerionen anbietet. Das Ausmaß der benötigten Technik für diese Therapien kann auszugsweise in Abbildung 2.2 erahnt werden. In der Abbildung ist lediglich die Vorrichtung zur Ablenkung des Strahls in die drei Therapieräume abgebildet. Hinzu kommen die riesige Gantry, der Synchrotron, und die drei Therapieräume in einem meterdick abgeschilderten, teils unterirdischen Bunker.



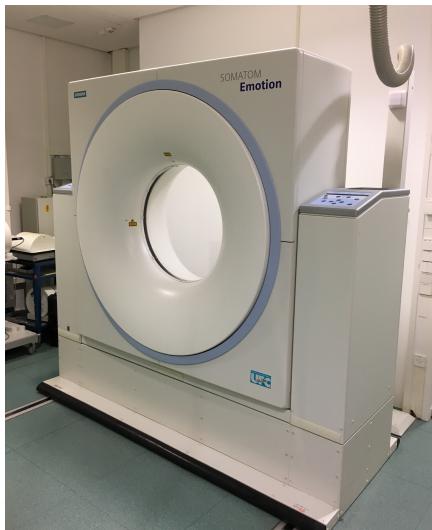
Abbildung 2.2: Nach dem Synchrotron teilt sich die Strahlenführung auf in Richtung der drei Behandlungsplätze. Das HIT verfügt über zwei horizontale Behandlungsplätze und einen Gantry-Behandlungsplatz. Zusätzlich gibt es einen Bestrahlungsplatz speziell für die Grundlagenforschung [Bild, Beschreibung: Uni12].

### 2.1.4 Bildgebung in der Strahlentherapie

In der Strahlentherapie wird meist mit Bildern der Computertomographie geplant. Diese wird in Abschnitt 2.1.4.1 vorgestellt. Neuerdings kommen auch vereinzelt Bildserien der Magnetresonanztomographie (MRT) zum Einsatz, welche in Abschnitt 2.1.4.2 näher erläutert wird. Bildgebung mit anderen Modalitäten, wie z.B. die funktionelle Magnetresonanztomographie (fMRT), Single Photon Emission Computed Tomography (SPECT), Positronen-Emissions-Tomographie (PET) oder die Skelettszintigraphie, werden hauptsächlich in der Diagnostik verwendet und finden in der Therapieplanung bei der Zielvolumendefinition Anwendung. Die Bildgebung mittels Ultraschall wird häufiger bei Tumoren im Bauchraum verwendet, um die Bestrahlungsplanung zu optimieren und den Therapieerfolg zu überprüfen [Ger17].

### 2.1.4.1 Computertomographie

Bei der Computertomographie (CT) werden Schnittbilder des Körpers mit Röntgenstrahlen erzeugt, wobei die Schwächung der Strahlen durch Dichteunterschiede der Gewebe nach dem Durchdringen durch den Patienten am Detektor gemessen wird. Ein Computertomograph besteht aus einer bewegbaren Patientenliege und einer rotierbaren Gantry in einem kippbaren Gehäuse (siehe Abbildung 2.3a). In der Gantry befinden sich Röntgenröhre, Blendensystem und Detektorsystem, die bei einer Aufnahme um den Patienten rotieren und die Schwächungen der Röntgenstrahlen hinter dem durchstrahlten Objekt als Schattenbilder auf dem Detektor erfassen. Mit diesen Schattenbildern kann die räumliche Verteilung der abschwächenden Gewebe entlang der Einstrahlrichtung am Computer rekonstruiert werden. Diese lassen einen Rückschluss auf die gewebespezifischen Schwächungswerte zu. Dadurch können bestimmte Gewebe hervorgehoben oder ausgeblendet werden [Kah17, S. 43-50] (siehe Abbildung 2.3b). Bei der Strahlentherapieplanung wird die Computertomographie eingesetzt, um durch die rekonstruierten Schwächungswerte der Gewebe deren Dichte und damit Absorptions-eigenschaft von Strahlung einzuschätzen. Die Computertomographie dauert nicht wesentlich länger als die Röntgenbildgebung. Der Patient wird allerdings mit mehr radioaktiver Strahlung belastet. Der Geräuschpegel durch die Kühlung des Geräts und die bewegende Gantry ist vernachlässigbar. Für die Planung der Strahlentherapie werden Lagerungshilfen, wie z.B. Vakuummatratzen oder ein Korsett angefertigt, um die Positionierung des Patienten am Linearbeschleuniger reproduzieren zu können.



(a) Computertomograph im DKFZ



(b) CT-Aufnahme eines Hirntumors [Gai17b]

Abbildung 2.3: Die Abbildungen zeigen einen Computertomographen im DKFZ in Heidelberg und eine Aufnahme eines Schädels mit Tumor in der rechten Hirnhälfte.

### 2.1.4.2 MR-Bildgebung

Die Magnetresonanztomographie (MRT) ist ein bildgebendes Verfahren, das ohne ionisierende Strahlung auskommt. Physikalisch basiert die Bildgebung auf den elektromagnetischen Eigenschaften von Wasserstoffkernen in organischen Verbindungen, welche durch Magnetfelder und hochfrequente Impulse angeregt werden, und deren minimale Magnetisierung anschließend gemessen wird. Die kompliziert gemessenen Rohdaten werden nach der Aufnahme zu 2D-Schichten und 3D-Volumina rekonstruiert. Anatomie kann mit der Magnetresonanztomographie je nach Gewichtung (z.B. T1, T2) bei der Messung mit sehr hohem Weichteilkontrast abgebildet werden, d.h. es ist eine feine Differenzierung von Geweben möglich [Sch12, S. 2] (siehe Abbildung 2.4). Ein Nachteil der Magnetresonanztomographie ist, dass die Aufnahme der Bildserien mit Aufnahmezeiten zwischen 15 Minuten bis zu über einer Stunde länger als z.B. bei der Computertomographie dauern. Dazu kommt, dass durch die hohe magnetische Anziehungschaft keine Metalle in die Nähe des Tomographen gebracht werden dürfen, da diese zu bedrohlichen Geschossen werden können und durch induzierte Wirbelströme zudem eine starke Erwärmung stattfinden kann. Der Patient wird in der häufig zwecks Homogenisierung des Magnetfelds engen Röhre des Tomographen positioniert, sodass bei einigen Patienten durch Platzangst Panik entstehen kann. Weiterhin entstehen durch das Schalten von Gradienten für das Erzeugen der verschiedenen Magnetfelder laute Klopf- und Brummgeräusche. Um die durchaus lange Untersuchung mit hoher Lautstärke erträglich zu machen, erhalten die Patienten Kopfhörer und eine Notfallklingel. Wie bei der CT-Bildgebung können Lagerungshilfen notwendig sein.

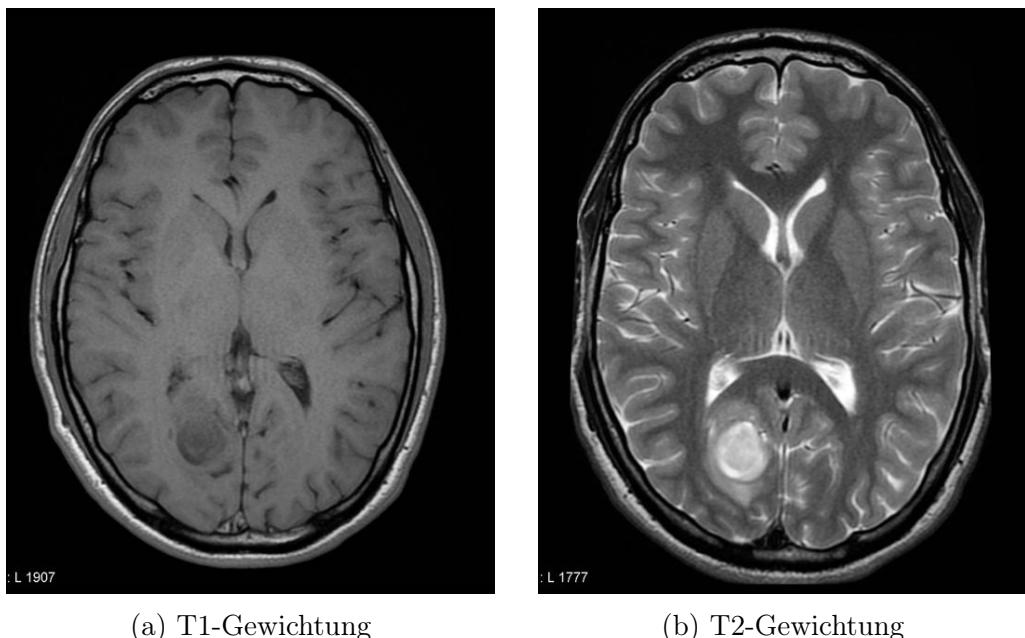


Abbildung 2.4: Tumor im rechten Okzipitallappen [Gai17a]

## 2.2 Aufklärung und Information von Patienten in der Strahlentherapie

Die nachfolgenden Abschnitte führen in die Aufklärung und Information von Patienten ein. Dabei werden schrittweise zunächst der Informationsbedarf und Angst auslösende Faktoren dargelegt. Anschließend wird auf verschiedene verwendete Medienformen eingegangen, um zu motivieren, wie diese aktuell sinnvoll eingesetzt werden. Jede dieser Medienarten alleine hat ihren Platz in der Information und Aufklärung. Dabei gibt es immer Faktoren, die beachtet werden müssen oder limitierend wirken können. Mit Hinblick auf den Einsatz von Virtueller Realität für die Information von Patienten, wird versucht herauszustellen, wie von diesen Erfahrungen profitiert werden kann.

### 2.2.1 Prozess der Therapie und Information des Patienten

Nach der Diagnostik und der Indikation einer Strahlentherapie findet mit dem behandelnden Strahlentherapeuten ein erstes Aufklärungsgespräch statt. Die diesem wird der Patienten ausführlich über Therapie, Risiken, Nebenwirkungen und das weitere Vorgehen informiert und besitzt die Möglichkeit, selbst Fragen zu stellen. Falls nötig, werden weitere Untersuchungen angeordnet. Zu jeder einzelnen Untersuchung wird der Patient gesondert aufgeklärt und erhält immer einen schriftlichen Aufklärungsbogen, in welchem der Ablauf der Untersuchung, alternative oder ergänzende Untersuchungen, Risiken, Komplikationen, wichtige Hinweise für die Zeit vor, während und nach der Untersuchung erläutert werden. Am Ende eines solchen Bogens werden mit spezifischen Fragen weitere Informationen eingeholt, um Gefahrenquellen, beispielsweise durch metallische Implantate bei indizierter MRT-Untersuchung, rechtzeitig zu erkennen. Nachdem alle notwendigen Informationen eingeholt wurden, wird ein Bestrahlungsplan erstellt. Häufig wird vor der Durchführung eines Planungs-CTs eine Lagerungshilfe angefertigt, um eine genaue Positionierung bei der Bestrahlung sicherzustellen. Am ersten Bestrahlungstermin wird der Patienten genauer über den konkreten Ablauf im Bestrahlungsraum mit Lagerung und Lagekontrolle, entsprechender Bildgebung und Abgleich mit dem Therapieplan informiert. Am letzten Tag der Bestrahlung findet ein Abschlussgespräch mit dem behandelnden Arzt statt, der den Patienten über den Erfolg der Therapie und das weitere Vorgehen informiert [Vgl. Med16].

### 2.2.2 Informationsbedarf der Patienten

Bereits Reynolds stellte 1978 fest, dass der Patient viel Information während der Therapie benötigte und folgerte daraus, dass keine Information stets schlechte Information sei [Rey78; Tho99a]. Fast alle Patienten möchten ausgiebige Informationen zu Therapie, Krankheit und Prognose [Mer96; Zeg12]. Es gibt jedoch auch Patienten, die wenig

Information über psychologische Hilfestellungen in Anspruch nehmen [Zeg12]. Dies wird durch die Ausführungen von Sutherland et al. unterstützt, welche besagen, dass Patienten aktiv nach Informationen suchen, um sich psychisch zu beruhigen [Sut89]. Butow et al. stellten fest, dass Informationsbedarf sowie aktive Beteiligung und Mithilfe der Patienten bei der Therapie sehr heterogen sind und nicht zwingend gemeinsam sinken und steigen müssen. Wenn sich die Prognose verschlechterte, wurden die Patienten passiver und ließen das medizinische Personal zunehmend selbstständiger handeln. Laut Timmermans et al. ist die Information und aktive Einbindung des Patienten gerade dann wichtig, wenn nur eine palliative Therapie möglich ist, damit kein Zurückziehen des Patienten erfolge [Tim05].

Nicht nur die Prognose hat Einfluss auf Informationsbedarf und aktive Beteiligung. Cassileth et al. stellten fest, dass auch das Alter des Patienten einen Einfluss auf die Arzt-Patientenrolle hatte: Je jünger der Patient war, desto eher wollte er ausgiebig informiert werden und bei der Therapie mitwirken. Je älter der Patient war, desto eher verblieb er in der Therapie passiv [Cas80; Fal95].

Die Information sollte mittels Papier als bevorzugtes Kommunikationswerkzeug erfolgen, da durch die vielen am Prozess beteiligten Mitwirkenden gegebenenfalls falsche oder widersprüchliche Informationen den Patienten verunsichern könne [Hin95; But98; Lon01; Rut05; Hal09]. Smith et al. identifizierten zwölf verschiedene Kategorien von Informationseinheiten, die für den Patienten während der Therapie von Relevanz sein könnten, wobei sich die behandelnden Institutionen, wie z.B. Krankenhäuser und Krebsgesellschaften, mit ihrem Informationsangebot über z.B. allgemeine Einschränkungen oder Auswirkungen auf das tägliche Leben ergänzen [Smi16]. Dennoch müsse man die individuelle Prognose des Patienten berücksichtigen, um einer überfordernden Informationsflut entgegenzuwirken [Smi16; Häg01; Zeg12].

### 2.2.3 Angst in der Therapie

Die Hälfte der in der Strahlentherapie behandelten Patienten leiden unter Angst [Els17]. Dabei korreliert eine höhere Grundangst mit hohem Informationsbedarf [Mes01], denn wenig Wissen über die komplexe Therapie führt zu Angst [Hal12]. Der Informationsbedarf muss identifiziert werden. Dazu gehören die Priorität der Information für jeden Patienten sowie die Informationsmedien und -quellen [Hal12]. Dass die Angst der Patienten nicht vernachlässigt werden darf, stellten Andersen et al. bei der Untersuchung dieses Sachverhalts mit gynäkologischen Krebspatientinnen vor. Dabei stiegen der Stress und die Verzweiflung der Patientinnen vor der ersten Behandlung. Bei der zweiten Sitzung erfolgte eine erneute Reaktion, die aber geringere Angst als vorher mit sich brachte [And84]. Zusätzlich seien laut Lamszus, Verres und Hubener die Bestrahlungssitzung, Wartezeit davor und Kontakt zu anderen Patienten stress- und angstfördernd [Lam94; Har93]. Anders als bei z.B. chirurgischen Eingriffen, führe die

Therapie, bedingt durch Fraktionierung, langanhaltende und regelmäßige Sitzungen, zu einer Steigerung der Ungewissheit gegen Ende der Therapie [And84]. Daraus ließe sich folgern, dass sich das Bedürfnis nach Information über die Zeit verändere und man den Patienten direkt nach Unklarheiten oder Sorgen fragen müsse [And85].

### 2.2.4 Einsatz und Effekt von Audioaufnahmen

Bei der Unterstützung der Information und Aufklärung der Patienten können zwei erlei Arten von Audioaufnahmen verwendet werden. Zum einen sind dies individuelle Mitschnitte der Aufklärungsgespräche, zum anderen standardisierte Aufnahmen allgemeiner Informationen. Reynolds et al. stellen fest, dass Audioaufnahmen helfen, die Kommunikation zwischen Arzt und Patient zu verbessern und Verständnis zu schaffen. Die Aufnahmen bilden zusätzlich ein Kommunikationsmedium für die Familie der Patienten [Rey81; Nor92]. Im Kontrast hierzu vertreten Knox et al. den Standpunkt, dass viele Patienten die Aufnahmen der Aufklärungsgespräche nicht möchten, da sie das Gespräch mit dem Arzt limitieren würden. Sie beobachteten, dass Patienten mit schlechter Prognose die Aufnahme vielmehr zur Information der ihnen Nahestehenden verwenden [Kno02]. Patienten, die die Aufnahme nach der ersten Behandlung angehört hatten, wussten laut Mood, Cook und Chadwell mehr über ihre Therapie [Moo88; For95; Hag96; Hac00]. Es bleibt jedoch zu beachten, dass der positive Effekt der Audioaufnahmen bei schlechter Prognose auch verschlimmern und angstfördernd wirken kann [McH95]. Weiterhin sollte, wenn die Aufnahme allgemeine Informationen enthält und nicht nur ein Mitschnitt eines Gesprächs ist, darauf geachtet werden, dass dieses auch den Abruf von Informationen behindern kann, da die Aufnahme vom Patienten als für den eigenen Fall nicht relevant angesehen werden könnte [Dun93]. Solche rein informativen Aufnahmen über die Verfahren der Therapie sind sinnvoll, wenn sie gezielt eingesetzt werden. Sie werden eher akzeptiert und verwendet, als Aufnahmen, welche die Gefühle der Patienten adressieren und Hilfestellungen zur Bewältigung der Angst bieten [Ben03].

### 2.2.5 Einsatz und Effekt von Videos und Simulationen

Videogestützte Aufklärung und Information erfolgen selten durch Videoaufnahme des Aufklärungsgesprächs, sondern mit standardisierten Informationen, die vorher mehr oder minder professionell erfasst wurden. Laut Thomas et al. finden etwa 80% diese informativen Aufnahmen sinnvoll [Tho99b]. Daran knüpfen sie die Bedingung, dass die Videos professionell, mit einer Filmfirma, produziert werden sollten [Tho00]. Harrison et al. relativieren dies, indem sie hervorheben, dass die Präsentation der Information unwichtiger als die Qualität der zu erlernenden Inhalte sei [Har00]. Dennoch gilt als unumstritten, dass Videos mit Informationsmaterial hilfreich und sinnvoll sind, da Patienten nachweislich gegenüber Patienten, die das Video nicht zusätzlich zu einer mündlichen Aufklärung erhielten, mehr Information abrufen können [Rai85; Hah05; Gon13; Wil17]. Denn das Video schaffe mehr Bewusstsein über die Relevanz von einzelnen Schritten der Verfahren und Therapie und führe zu einer verbesserten Mitwirkung des Patienten [Gon13; Wil17]. An Stelle von Videos werden auch Simulationen und interaktive Modelle verwendet, um den Patienten mehr Wissen über die Verfahren und Geräte zu vermitteln [Oya97; McD01; Hug15]. Dieser Effekt ist im Patientenkollektiv unterschiedlich ausgeprägt, wie Hahn et al. herausfanden. Denn Patientinnen mit Brustkrebs fanden die Videos statistisch signifikant besser, als Patienten mit anderen Krebsarten [Hah05]. Videos können jedoch auch eine andere Rolle spielen, wie Clotfelter zeigt. Er wies nach, dass ältere Leute ein verbessertes Schmerzmanagement zeigen und Videos ihren Umgang mit Schmerzen im Alltag verbessern [Clo99]. Eine andere Art und Weise der Nutzung zeigen Shrimpton et al. Sie ließen krebskranke Kinder selbst Videos in der Form eines Tagebuchs drehen. Dabei zeigten sie, dass die Videos hilfreich für die Wiedereingliederung in den Alltag und Erklärung der Therapie gegenüber Gleichaltrigen war. Zudem wiesen die Kinder durch die aktive Beschäftigung mit der Materie reduzierte Vorbehalte gegenüber der Therapie und Krankheit auf [Shr12].

## 2.3 Virtuelle Realität

Dieser Abschnitt behandelt die Virtuelle Realität, auch Virtual Reality (VR) genannt, und ihre Konzepte. Zunächst wird die Idee hinter Virtual Reality und anschließend die Prinzipien der Virtuellen Realität vorgestellt. Dabei soll motiviert werden, durch welche Konzepte eine Illusion erreicht werden kann, die für die Aufklärung und Information von Patienten verwendet werden kann, um diese in Situationen versetzen zu können. Gleichzeitig wird versucht dem Leser einen Eindruck der Komplexität und des Aufwands zu vermitteln, der mit dem Erzeugen dieser Illusionen erbracht werden muss.

Bereits 1965 beschrieb Ivan E. Sutherland die Kerneigenschaft der meisten heutzutage entwickelten VR Anwendungen. Die beschriebene Virtuelle Welt wirkt überzeugend und nutzt die Erwartungen und Sinne des Menschen, um eine *perfekte* Benutzerschnittstelle zu bilden, wie im Folgenden Zitat deutlich wird:

“The ultimate display would, of course, be a room within which the computer can control the existence of matter. A chair displayed in such a room would be good enough to sit in. Handcuffs displayed in such a room would be confining, and a bullet displayed in such a room would be fatal. With appropriate programming such a display could literally be the Wonderland into which Alice walked.”

— Ivan E. Sutherland, 1965 [Sut65]

Diese Vision fordert eine Illusion, welche die Realität komplett durch eine Virtualität ersetzt. Das ist in der heutigen Umsetzung der Virtuellen Realität nur begrenzt realisiert und gewollt. Es wird von einer Virtuellen Umgebung gesprochen, die durch verschiedene Illusionen und technische Ausstattungen versucht, ausgewählte Aspekte möglichst real wirken zu lassen. Dabei werden bewusst einige, häufig nachteilige, Aspekte der Realität weggelassen.

### 2.3.1 Virtuelle Umgebung

Ein VR-System wird nach Dörner als ein System aus geeigneter Hardware und Software, die im Zusammenspiel die Vorstellung einer Virtuellen Welt realisiert, beschrieben. Dabei definiert er die im VR-System dargestellten Objektmodelle, deren Verhalten und ihre Anordnung in der Szene als Virtuelle Welt. Die mit dem VR-System dargestellte Virtuelle Welt nennt er Virtuelle Umgebung [Dör13, S. 7]. In der Literatur wird diese häufig mit dem englischen Begriff Virtual Environment (VE) bezeichnet. Bricken definiert dies sogar als das wesentliche Merkmal der Virtuellen Realität [Bri90].

### 2.3.2 Präsenz

Das Gefühl, sich in einem Virtual Environment zu befinden, wird Präsenz bzw. Telepräsenz genannt und macht das Hauptmerkmal der Virtuellen Realität aus [Dör13, S. 18]. Die Darstellung der Virtuellen Welt allein reicht jedoch nicht aus. Slater gliedert dazu die Präsenz in zwei Teilespekte auf, die Ortsillusion und die Plausibilitätsillusion [Sla09, S. 1].

#### 2.3.2.1 Ortsillusion und Virtual Body

Die Ortsillusion fordert, dass die Präsentation der 3D-Inhalte zusätzlich zum Sehsinn auch mit weiteren Sinnen, wie z.B. dem Hörsinn oder Tastsinn erfolgen muss [Dör13, S. 13]. Darüber hinaus muss der Nutzer laut Gutiérrez et al. die Möglichkeit haben, sich in der Szene frei zu bewegen und ihre Merkmale zu erkunden, auch Navigation genannt [Gut08, S. 1]. Das Drehen des Kopfes soll zu einer entsprechend veränderten Anzeige auf dem Display führen und gegebenenfalls die Richtung der Audioquelle verändern, sodass diese im Raum invariant zur Kopfbewegung des Nutzers ist. Dies macht es notwendig, dass Körperbewegung und Körperlage im Raum oder die Lage einzelner Körperteile zueinander vom System erfasst werden und entsprechendes Feedback generiert werden muss [Sla97, S. 3]. Dafür wird sogenanntes Tracking verwendet, d.h. ein Verfolgen der Position und Orientierung von Gegenständen im dreidimensionalen Raum. Damit das Feedback des Systems natürlich wirkt, wird eine Selbstdarstellung des Nutzers oder einzelner Körperteile im Virtual Environment notwendig, der sogenannte Virtual Body (VB). All diese Forderungen führen dazu, dass der Nutzer sich an einen anderen Ort versetzt fühlt, obwohl er sicher weiß, dass er dort nicht ist.

#### 2.3.2.2 Plausibilitätsillusion

Plausibilitätsillusion ist die Illusion, dass das präsentierte Geschehen wirklich passiert ist. Eine Schlüsselkomponente der Plausibilitätsillusion ist, dass Ereignisse in der Virtuellen Umgebung, über die der Nutzer keine direkte Kontrolle hat, direkt auf ihn beziehen, wie z.B. wenn er von einer virtuellen Person direkt angesprochen wird. [Sla09, S. 8]

### 2.3.3 Immersion

“Immersion ist eine Beschreibung einer Technologie und beschreibt das Ausmaß, in dem die Computer-Displays in der Lage sind, den Sinnen eines menschlichen Teilnehmers eine eingeschlossene, umfassende, umgebende und lebendige Illusion der Wirklichkeit zu liefern.” [Eigene Übersetzung Sla97, S. 3] Dabei meinen Slater und Wilbur den Grad, zu welchem die physische Realität ausgeschlossen ist, wie viele verschiedene

Sinne angesprochen werden, die Breite des Sichtfelds und des Informationsgehalts, der Auflösung und der Qualität der Displays [Sla97, S. 3].

### 2.3.4 Willentliches Ausblenden des Unglaubens

“Menschen haben die Eigenschaft in bestimmten Situationen den augenscheinlichen Widerspruch einer virtuellen oder fiktiven Welt zur Realität auszublenden und dies auch zu wollen.” [Dör13, S. 8] Dies wird auch mit dem englischen Ausdruck *willing suspension of disbelief*, seltener auf deutsch *willentliches Ausblenden des Unglaubens*, ausgedrückt. Dies wird in Anwendungen der Virtuellen Realität ausgenutzt, weshalb geringfügige Abweichungen von der Realität vorhanden sein können, ohne das Erlebnis der Präsenz zu schmälern.

## 2.4 Augmented Reality (AR)

Augmented Reality (AR), auch erweiterte Realität genannt, basiert auf der Erweiterung der Umgebung mit virtuellen Objekten, sodass diese miteinander zusammenwirken, um einen einheitlichen Eindruck zu erwecken. Der Eindruck wird dadurch erweckt, dass die virtuellen Objekte abhängig vom Blickwinkel in die Realität eingebettet werden, sodass die Position und Orientierung im Raum immer gleich bleibt. Dazu wird mit einer Kamera kontinuierlich ein Video aufgenommen. Durch dieses Video wird die Bewegung der Kamera im Raum verfolgt, woraus geschätzt wird, wie die realen Objekte im Raum zueinander und zur Kamera positioniert und orientiert sind. Anschließend wird das virtuelle Objekt in dieser Perspektive registriert, d.h. positioniert, skaliert und orientiert. Abschließend wird das Kamerabild samt virtuellem Objekt auf einem Display ausgegeben [Vgl. Dör13, Kapitel 8]. Augmented Reality ist einfacher zu erreichen als Virtuelle Realität, da man die Umgebung nicht komplett durch eine Virtuelle Umgebung ersetzt, sondern diese durch virtuelle Objekte ergänzt, was bereits mit der Kamera und Rechenpower eines Smartphones umsetzbar ist. In Abbildung 2.5 ist ein AR-Spiel dargestellt, bei welchem man virtuelle Bauklötze stapeln muss. Der Turm bleibt in dem Spiel immer an derselben Stelle im Raum stehen. Werden die Bauklötze nicht sauber aufeinander gelegt, so fällt der überhängende Teil des Klotzes auf den Boden. Dabei wird die Physik des Klotzes berechnet, sodass dieser möglichst real beschleunigt und bei Kollision mit anderen Bauklötzen und dem Boden ein reales Streuen simuliert wird.



Abbildung 2.5: Stack AR als Beispiel für eine Augmented Reality Anwendung. In dem Spiel müssen Bauklötzte möglichst deckend gestapelt werden.

## 2.5 Einsatz von VR & AR in der Medizin

Virtual Reality wird in der Medizin in vielen Bereichen verwendet. Vor allem in der Chirurgie werden bei der Ausbildung von Medizinern Abläufe in Operationen [Kho16], wie z.B. laparoskopische Eingriffe [Nic15] oder Eingriffe in der Neurochirurgie [Ala11] durch virtuelle Systeme simuliert. Andere Einsatzgebiete finden sich in der Konfrontationstherapie, wenn Patienten mit z.B. Höhenangst [Rot95] gezielt Angst auslösenden Reizen und Umgebungen ausgesetzt werden. In der Chemotherapie wird vereinzelt versucht, den Patienten während der Infusion des Chemotherapeutikums von der Therapie abzulenken [Sch03; Sch07]. Augmented Reality findet bereits großen Einsatz in der Chirurgie, wenn anatomische Strukturen während der Operation in einem Mikroskop oder Tablet [Tan17] überblendet werden. Ein anderes Einsatzgebiet ist die Visualisierung von mittels Bildgebung erfasster Anatomie in der Forensik [Kil15]. Bei den meisten dieser Anwendungen ist der Nutzer Teil des medizinischen Personals, sodass Information, die bereits vorhanden ist, alternativ dargestellt wird, um den Therapie- oder Untersuchungsprozess zu unterstützen. Häufig entsteht der Patientenbezug nach einer Anwendung des medizinischen Personals, sodass die Software um diese Komponente erweitert wird.

## 2.6 Spieleentwicklung mit Game Engines

In diesem Abschnitt werden zunächst die Grundlagen einer 3D-Szene vorgestellt (Abschnitt 2.6.2). Anschließend werden die Werkzeuge dargelegt, die benötigt werden, um eine statische Anzeige einer Szene zu einer dynamisch verändernden Szene zu machen. Im Anschluss wird vorgestellt, was der Vorteil von sogenannten Game Engines ist und welche Limitationen diese in der Regel mit sich bringen. Dieses Grundlagenwissen soll dafür sorgen, dass die im Methoden- und Ergebnis-Kapitel beschriebenen Verfahren und Konzepte verstanden werden. Game Engines spielen im medizinischen Kontext dieser Arbeit eine Rolle, da sie die Entwicklung einer VR-Anwendung stark vereinfachen und so zu schnelleren und ausgereifteren Ergebnissen führen.

### 2.6.1 3D-Objekte aus Vertices, Edges, Faces und Normalen

Damit ein Objekt in einer 3D-Szene dargestellt werden kann, muss es digital repräsentiert werden. Jedes Objekt besitzt dabei eine Oberfläche, die in einem kartesischen Koordinatensystem platziert ist, welches durch dessen Ursprung repräsentiert wird. Das Objekt besteht primär aus Punkten, sogenannten Vertex Points, die relativ zum Ursprung des Koordinatensystems platziert sind. Die Vertices sind untereinander mit Kanten, den sogenannten Edges, verbunden. Dadurch entsteht ein Gittermodell des Objekts, auch Wireframe genannt. In Abbildung 2.6 ist ein Wireframe eines Würfels mit dem Ursprung des Koordinatensystems sowie den Koordinaten der Vertices (grau) dargestellt. Damit das Objekt eine visuelle Repräsentation erhält, muss die Oberfläche, auch Mesh genannt, definiert werden. Dafür bilden Polygone, wie z.B. Dreiecke oder Vierecke, eine Fläche, auch Face genannt. Zusätzlich dazu kann für jedes Face angegeben werden, welches die Außen- und Innenseite ist. Das wird durch eine Normale bewerkstelligt, die lotrecht auf dem Face steht. Dabei gibt es die Möglichkeit, für jeden Vertex und oder die Faces Normalen zu definieren. Die Faces können gefärbt oder mit einer Textur belegt werden [Fla10]. Speichert man ein solches Objekt, so entstehen meist drei Dateien, die gemeinsam das Objekt und seine visuelle Repräsentation bilden. Das sind zum einen eine `obj`-Datei, die alle Punkte mit ihren Koordinaten relativ zum Ursprung enthält, sowie alle Normalen, die durch Richtungsvektoren repräsentiert werden. An diese Liste folgt eine Zuordnung der Vertices und Normalen zu Kanten und Faces sowie Texturkoordinaten durch Zusammenfassen der Indices. Die zweite Datei definiert das Material der Oberfläche und gibt beispielsweise Transparenz und Streuung von Licht auf der Oberfläche an. Die dritte Datei ist ein Bild der Textur der Oberfläche, aus welcher die Faces mit ihren Texturkoordinaten die entsprechenden Bildausschnitte beziehen.

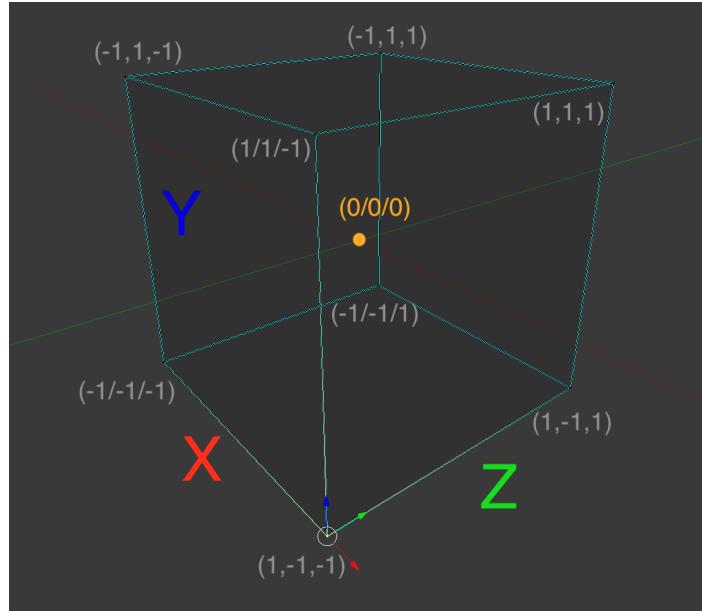


Abbildung 2.6: 3D-Wireframe eines Würfels mit Vertices (Knoten) und Edges (Kanten). Die Koordinaten der Vertices werden relativ zum Ursprung des Objektkoordinatensystems angegeben.

### 2.6.2 Aufbau einer 3D-Szene

Eine 3D-Szene ist in einem Szenengraphen organisiert. Ein Beispiel für einen Szenengraphen ist in Abbildung 2.7 dargestellt. Der Szenengraph ist prinzipiell ein Baum, dessen Wurzelknoten das Universum ist, d.h. der Raum in dem die Objekte platziert werden. An diese Wurzel werden alle anderen Knoten direkt oder indirekt angehängt. Die Knoten repräsentieren dabei in der Szene platzierte Objekte, Transformationen, Kameraposen, Animationen und viele weitere mögliche Inhalte einer 3D-Szene.

Häufig besteht ein in der Szene dargestelltes Objekt aus mehreren Einzelteilen die relativ zueinander positioniert sind. In der Abbildung ist dies das Objekt Fahrrad, das aus zwei Rädern und einem Rahmen besteht. Jedes dieser Objekte besitzt eine Transformation absolut zum Weltkoordinatensystem der Szene. Damit bei Transformationen, wie z.B. Rotationen oder Skalierungen, nicht für jedes Objekt einzeln die Transformationen neu berechnet werden müssen, werden diese ebenfalls in einer Hierarchie angeordnet. Das bringt den Vorteil, dass die Teile zur Wurzel dieser Objekthierarchie relativ angeordnet werden können. Da Transformationen in der Regel durch Transformationsmatrizen verarbeitet werden, bringt dies den Vorteil, dass lediglich die Wurzel bei der Transformation absolut transformiert werden muss. Anschließend kann für alle relativ dazu positionierten Knoten des Objekts durch Matrixmultiplikation mit den Koordinaten der Wurzel und der Transformationsmatrix der relativen Positionierung

die neue Position berechnet werden. In der Abbildung ist dies daran zu erkennen, dass die Translation der Räder des Fahrrads relativ der Transformation des Fahrrads untergeordnet ist. Der Szenegraph ist ein gerichteter azyklischer Graph und kein Baum, weil zusätzlich zu der hierarchischen Anordnung auch Referenzen zu anderen Teilbäumen erstellt werden können. In der Abbildung ist dies beispielsweise bei der Animation des Fahrrads veranschaulicht. Ein Beispiel dafür ist die Instanziierung mehrerer gleichartiger Objekte in der Szene. Die Zusammensetzung des Objekts ist in eine Teilbaum mit der relativen Anordnung der Einzelteile organisiert. In einem anderen Teilbaum wird die absolute Transformation des Objekts im Raum definiert.

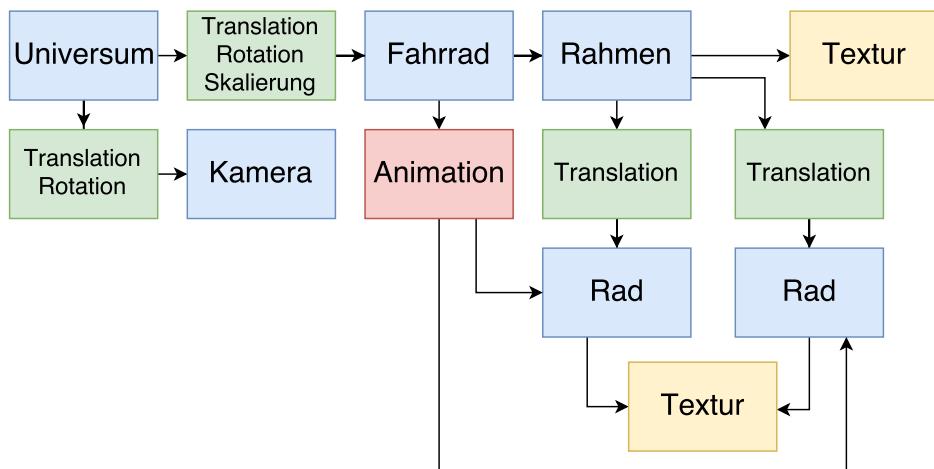


Abbildung 2.7: In der Abbildung ist der Aufbau eines Szenegraphen mit Universum und einem Fahrrad abgebildet. Das Fahrrad besteht aus einem Rahmen und zwei Rädern, die relativ zum Rahmen verschoben sind. Die Räder sowie der Rahmen besitzen eine Textur. Eine Animation des Fahrrads bezieht sich auf die Räder. Die Szene wird durch eine im Raum positionierte und orientierte Kamera betrachtet. Das Fahrrad selbst ist in der Szene skaliert, positioniert und orientiert.

### 2.6.3 Inhalte einer 3D-Szene

Im Szenegraphen einer 3D-Szene sind verschiedene Typen von Objekten organisiert. Die am häufigsten verwendeten Typen sind Meshes mit ihren Texturen. Damit die Objekte sichtbar sind, muss für Beleuchtung gesorgt werden. Dabei gibt es in der Regel drei verschiedene Arten - Static Light, Point Light und Directed Light. Static Light ist die Grundbeleuchtung einer Szene, welche unabhängig von der Distanz zur Quelle gleich stark beleuchtet, wie es beispielsweise durch die Sonne erreicht wird. Das Point Light ist ein Licht, das punktförmig radial in alle Raumrichtungen mit einer Intensität, die umgekehrt quadratisch abhängig zur Distanz Licht emittiert. Dies

ist mit einer Glühbirne vergleichbar. Das Directed Light kann mit einem Scheinwerfer verglichen werden, da ein Lichtkegel ebenfalls mit einer Intensität, die umgekehrt quadratisch abhängig zur Distanz zur Quelle ist, Licht emittiert, im Gegensatz zum Point Light jedoch in ihren Ausbreitungsrichtungen durch den definierten Lichtkegel beschränkt ist. Für jede dieser Arten von Licht können verschiedene Eigenschaften, wie z.B. Schattenwurf, Farbe oder Intensität, definiert werden. Die selben drei Unterscheidungen werden bei in der Welt platzierten Audiospuren vorgenommen, da diese immer gleich laut im Hintergrund, punktförmig schallend oder gerichtet, wie ein Lautsprecher, sein können. Es gibt viele weitere Dinge, die in einer 3D-Szene definiert sein können, wie z.B. Animationen, Charaktere, Navigationsmeshes und Kollisionsmeshes. Navigationsmeshes geben an, an welchen Orten einer Szene ein Charakter positioniert werden kann. Kollisionsmeshes sind stark vereinfachte Oberflächenrepräsentationen von Objekten, die für die Berechnung von Kollisionen verwendet werden, da eine Berechnung mit der originalen Oberfläche rechnerisch zu komplex ist. Eine 3D-Szene besitzt zusätzlich zu allen Objekten eine zugrundeliegende Physiksimulation. Diese spezifiziert, wie Kollisionen und Objektbeschleunigungen ablaufen oder wie stark Anziehungskräfte, wie z.B. die Gravitation, sind.

### 2.6.4 Game Engines

In der Spieleentwicklung haben sich Frameworks, d.h. Programmiergerüste, gebildet, die die komplexe Entwicklung eines Spiels vereinfachen sollen, indem sie häufig benötigte Werkzeuge bündeln. Diese sogenannten Game Engines bringen häufig eine auf sie zugeschnittene Entwicklungsumgebung mit. Eine Game Engine besteht aus vielen einzelnen Komponenten. Die wichtigste Komponente ist die Grafik-Engine, die für die grafische Darstellung, d.h. das Rendering der Meshes und ihrer Texturen sorgt. Weitere Komponenten sind die Physik-Engine sowie das Soundsystem. Hinzu kommen häufig einige weitere Komponenten für die Netzwerkkommunikation, ein System für das Verwalten des Zustandsautomaten des Spiels und Speichern des Spielstands. Weitere Systeme übernehmen die Verwaltung der Steuerung mittels z.B. einer Tastatur, Maus oder einem Gamepad. Die Editoren der Game Engines bieten weitreichende spezialisierte Werkzeuge für das Erstellen und Manipulieren aller Inhalte einer 3D-Szene. Die Schnittstellen dieser Werkzeuge werden über eine komplexe Programmierschnittstelle in einer Programmiersprache, wie z.B. C++ bereitgestellt.

## 3 Material und Methoden

In diesem Kapitel wird zunächst auf Verfahren zur Digitalisierung von realen Objekten eingegangen, da das Ziel dieser Arbeit die Entwicklung eines Verfahrens einschließt, um die z.B. in Kapitel 2.1 vorgestellten Geräte in eine VR-Anwendung zu integrieren. Anschließend werden die für dieses Verfahren verwendete Software (3.2) und Algorithmen (3.3) erläutert. In Abschnitt 3.5 werden die zwei größten Game Engines für die Entwicklung des Demonstrators vorgestellt.

### 3.1 Verfahren zur Digitalisierung realer Objekte

In Abschnitt 3.1.1 wird erläutert, wie man durch den Ansatz des 3D-Scannens Objekte automatisiert erfassen kann. Im darauffolgenden Abschnitt 3.1.2 wird vorgestellt, wie manuell Objekte in 3D modelliert werden können.

#### 3.1.1 3D-Scannen

Das Prinzip des 3D-Scannens liegt darin, dass ein Objekt aus verschiedenen Richtungen durch z.B. einen Laser abgetastet wird. Das bedeutet, dass das Objekt angestrahlt und das reflektierte Licht als Signal gemessen wird. Je nach Technik wird dies durch Licht-Pulse und die Messung der benötigten Zeit bis der Puls einen Sensor erreicht, Time of Flight (ToF) genannt, oder durch Bestrahlen mit einem Muster durch einen Projektor und Messung der Verzerrung dieses Musters durch Kameras, Structured Light genannt, erreicht. Je nach Größe des Objekts muss dafür eine andere Technik verwendet werden. Professionelle Laser-Scanner, die für die Vermessung von Objekten in der Bauindustrie verwendet werden, tasten mit dem Laser große Flächen und Volumen ab. Anschließend wird der Laser-Scanner neu positioniert, sodass dieser das Volumen aus möglichst vielen verschiedenen nachvollziehbaren Positionen erfassen kann und die Daten am Computer zusammengeführt. Für die Vermessung eines kleinen Objekts ist diese Technologie ungeeignet. Die Scans müssen eine hohe Auflösung besitzen, damit auch feine Strukturen erfasst werden können. Dafür gibt es kleinere Scanner, die mit der Hand um das Objekt geführt werden können. Die Grundlage dieser Scanner ist es immer, den Abstand des Scanners zum Objekt zu messen und daraus zu berechnen, wie die gemessenen Punkte zueinander positioniert sind. Große Areale werden, wie z.B. bei der Photogrammetrie, mit Flugobjekten, wie z.B. Drohnen, erfasst. Dabei wird die Bewegung des Flugobjekts zusätzlich zu 2D-Aufnahmen des Areals erfasst und bei einem Bündelblockausgleich (Abschnitt 3.3.1) ein Modell parametrisiert. Ein Problem des 3D-Scannens ist, dass durch Rauschen und Artefakte bei der Erfassung des Objekts häufig eine aufwändige nachträgliche Bearbeitung des Modells notwendig ist.

### 3.1.2 Modellierung von 3D-Objekten

Für die Modellierung von 3D-Objekten wird eine Modellierungssoftware, wie z.B. Blender (Abschnitt 3.2.1), benötigt. Diese Software stellt Werkzeuge bereit, um über verschiedene Ansichten des Objekts schrittweise die Oberfläche des Objekts mit Standardgeometrien, wie z.B. Quadraten und Dreiecken, zu modellieren. In Abbildung 3.1 ist dargestellt, wie man durch schrittweises Erweitern einer Oberfläche in 2D (links) und anschließendes Verschieben der Knotenpunkte in der noch fehlenden dritten Dimension (rechts) eine 3D-Oberfläche modelliert. Dafür werden frontale und seitliche Aufnahmen des zu modellierenden Objekts benötigt, welche in der Modellierungssoftware hinter die Arbeitsfläche gelegt werden können. Die Erweiterung der Oberfläche kann beschleunigt werden, wenn man symmetrische Oberflächen an der Symmetrieebene spiegeln lässt. Durch Rotation, Translation und Skalierung kann so schrittweise ein 3D-Modell erstellt werden. Nach der Modellierung kann man die modellierte Oberfläche durch die Modellierungssoftware und vorhandene Fotos des realen Objekts texturieren. Das heißt, jedem Polygon oder Dreieck der Oberfläche wird eine eindeutige Abbildung auf die Fotos zugeordnet. Sogenannte CAD-Software ermöglicht das millimetergenaue Modellieren von z.B. mechanischen Modellen oder Maschinen.

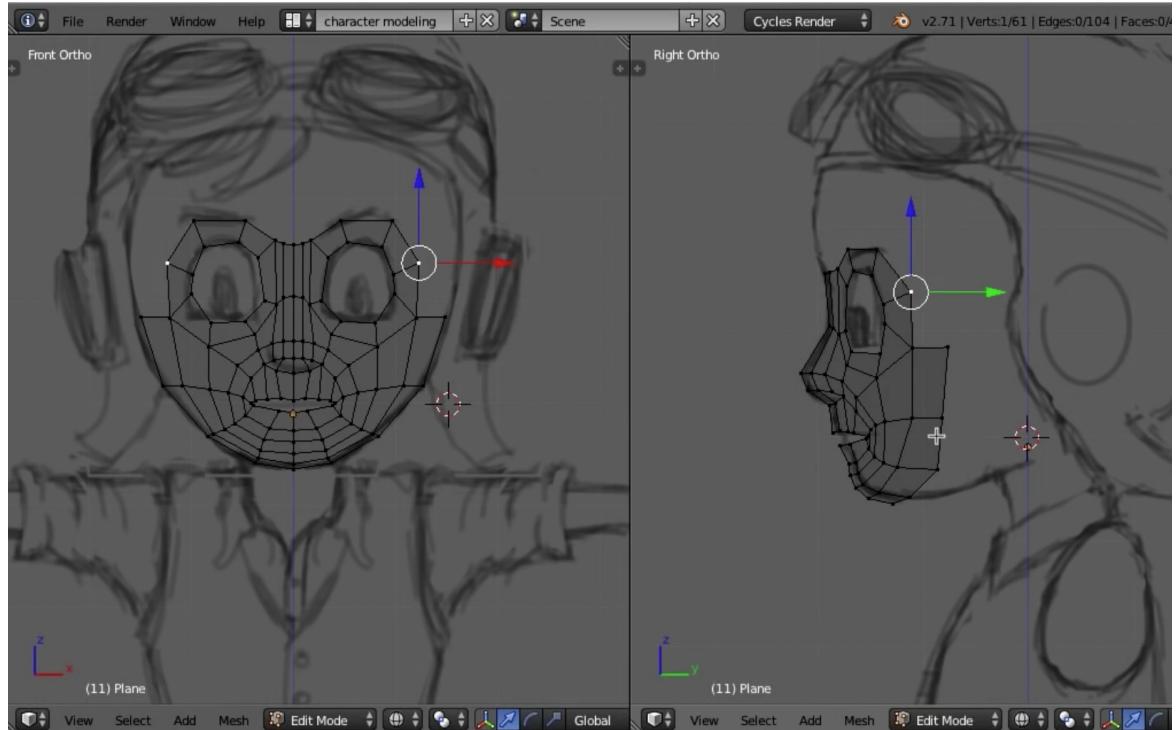


Abbildung 3.1: Modellieren eines 3D-Charakters mit Blender [Bildausschnitt aus: Lil14]. Mit der Software kann schrittweise das Oberflächennetz erweitert werden. Durch Symmetrieeachsen und verschiedene Ansichten entsteht eine 3D-Oberfläche.

## 3.2 Software der Digitalisierung realer Objekte

Für die Digitalisierung realer Objekte gibt es viele kommerzielle und frei verfügbare Software. In den folgenden Unterabschnitten werden drei Anwendungen vorgestellt, die zum Modellieren (3.2.1), 3D-Scannen (3.2.2) und einer eventuellen Korrektur (3.2.3) der Modelle verwendet werden.

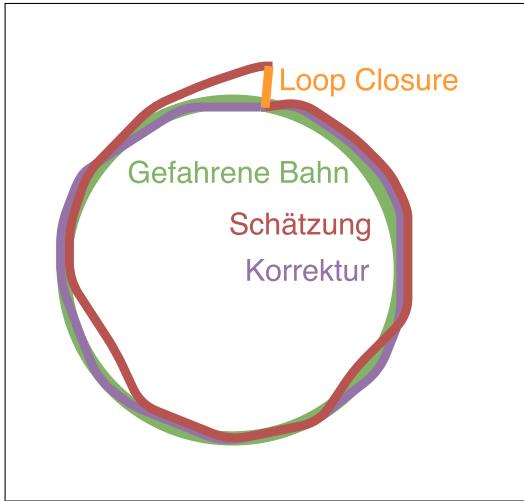
### 3.2.1 Blender

Blender ist die Open-Source 3D-Modellierungssoftware. Die Anwendung beinhaltet viele Werkzeuge zur 3D-Modellierung, Texturierung, Animation, Simulation, dem Rendering, der Videobearbeitung und Spielerstellung. Für fortgeschrittene Benutzer gibt es ein Python-Application Programming Interface (API), um die Anwendung anzupassen und spezialisierte Werkzeuge zu schreiben [Sti17].

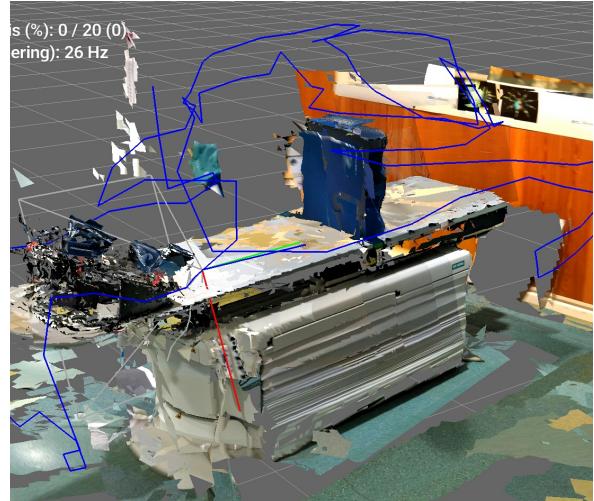
### 3.2.2 RTAB-Map

RTAB-Map ist eine Open-Source Software für die Rekonstruktion von 3D-Szenen, des IntRoLab, einem Robotiklabor der Universität Sherbrooke in Kanada, die nach dem gleichnamigen Algorithmus [Lab13] benannt wurde. RTAB-Map steht für Real-Time Appearance-Based Mapping. Dieser Farb- und Tiefeninformation (RGB-D) verarbeitende Simultaneous Localization And Mapping (SLAM) Ansatz konstruiert dynamisch eine Karte der aufgenommenen Umgebung auf Basis eines Graphen. Die Knoten des Graphen repräsentieren die Posen der Kamera auf jeder Position der Karte. Als SLAM wird in der Robotik das Problem des Anlegens einer Umgebungskarte eines Roboters und das gleichzeitige Schätzen seiner Position und Orientierung in dieser Karte bezeichnet. Zusätzlich zu dieser Information beinhalten die Knoten Informationen zur Visualisierung, wie z.B. Laser Scans, RGB-Bilder, Tiefeninformation und Features für die Loop Closure Detection. Die Kanten des Graphen stellen die rigiden geometrischen Transformationen zwischen den Knoten dar [Lab14]. Als eine Loop Closure Detection bezeichnet man ein Verfahren, das die Robustheit eines Simultaneous Localization And Mapping (SLAM) Algorithmus durch Erkennen von bereits besuchten Orten erhöht. Für RTAB-Map bedeutet dies, dass beim Umlaufen von Objekten während eines Scans mehrfach gescannte Oberflächen durch die Loop Closure Detection erkannt und zusammengeführt werden. In Abbildung 3.2a ist am Beispiel eines Kreises dargestellt, wie der in grün markierte Kreis beim Scan mit einem Tablet abgelaufen und geschätzt wurde. Hierbei treten Ungenauigkeiten auf, sodass die aufgenommene Karte an der Ausgangsposition nicht deckend ist. Durch die Loop Closure Detection kann diese Abweichung erkannt werden (orange dargestellt) und die Karte nachträglich korrigiert werden (lila). Die Software ist für verschiedene Plattformen online frei verfügbar

[RTAB]. Eine Android-App ermöglicht das Scannen mit Geräten von Google's Project Tango, wie z.B. Lenovo's Phab 2 Pro, welche spezielle Hardware für die Messung der Tiefe zu einem Objekt besitzen. Das Scannen mit dem Tablet ist in Abbildung 3.2b dargestellt. Der Bewegungspfad des Tablets im Raum ist mit einer blauen Linie dargestellt. Die Anwendung rekonstruiert die gescannten Punktwolken durchgängig mit niedriger Qualität, um visuelles Feedback über die bereits erfassten Strukturen zu geben. Die aufgenommenen Daten können in gängige 3D-Formate exportiert werden. Eine Desktop-Anwendung ermöglicht das Rekonstruieren an einem leistungsfähigeren Computer, um noch genauere und schnellere Ergebnisse zu erhalten.



(a) Die Loop Closure Detection der Positions- und Orientierungsschätzung (rot) erhöht die Robustheit der Schätzung der Bahn (grün) durch das Erkennen von Schleifen (orange), sodass Ungenauigkeiten korrigiert werden können (lila).



(b) Ausschnitt der 3D-Visualisierung eines Scans mit der RTAB-Map App. Die gescannten Punktwolken werden durchgängig mit niedriger Qualität rekonstruiert, um visuelles Feedback über die bereits erfassten Strukturen, sowie den Bewegungspfad (blau) zu geben.

Abbildung 3.2: RTAB-Map legt beim SLAM eine Umgebungskarte an, welche durch einen Loop Closure Detection Ansatz kontinuierlich korrigiert und entlang des Bewegungspfads des Tablets im Raum erweitert werden. Die Anwendung gibt live visuelles Feedback über den Scan. Durch die Karte und erfasste Punktwolken mittels einer Tieffenkamera kann ein 3D-Modell berechnet werden.

#### 3.2.3 MeshLab

Meshlab ist eine Open-Source-Software für die Verarbeitung und Bearbeitung von 3D-Oberflächennetzen und Punktwolken. Sie stellt eine Reihe von Werkzeugen für die Bearbeitung, Reinigung, Reparatur, Inspektion, das Rendern, die Texturierung und Konvertierung von Oberflächennetzen zur Verfügung [Cig08] (siehe Abbildung 3.2). Sie wird von dem Visual Computing Lab entwickelt, einem Labor des Institut für Informationswissenschaft und Technologie (ISTI) des Nationaler Forschungsrat (CNR) in Pisa, Italien. Die Software kann verwendet werden, um zahlreiche gängige Formate von Punktwolken und Oberflächengittern sowie deren Texturen zu laden und zu verarbeiten. Sie bietet Funktionen für die Nachbearbeitung bereits triangulierter Oberflächen, z.B. die Erkennung und Entfernung von doppelten, nicht referenzierten Punkten, kleineren isolierten Flächen, physikalisch unmöglichen Geometrien oder zum Schließen von kleineren Löchern in Oberflächen. Andere beinhaltete Werkzeuge umfassen Algorithmen zur Reduzierung der Anzahl von Flächen, Oberflächenrekonstruktionen von Punktwolken, z.B. die Poisson Surface Reconstruction (PSR), merkmalserhaltende Glättung und Farb-Matching-Filter [Cig08].

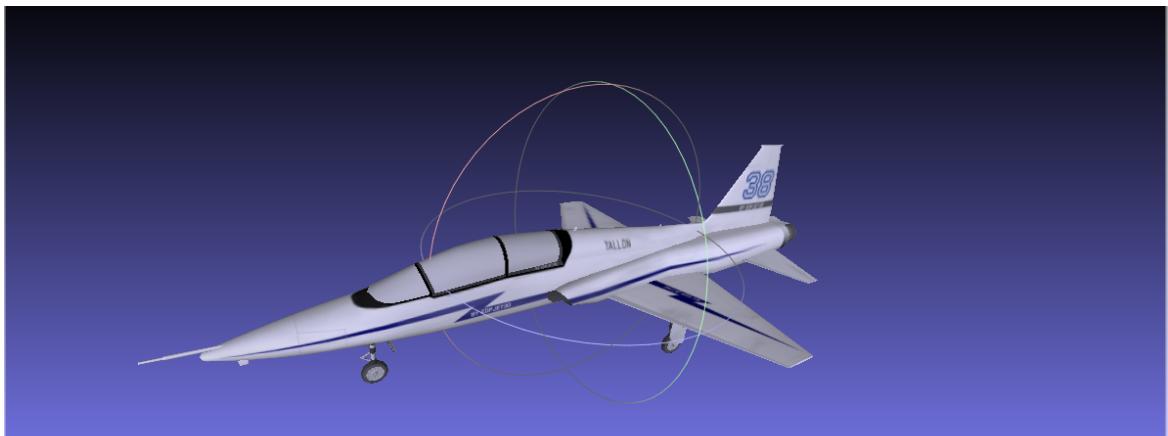


Abbildung 3.3: Darstellung eines 3D-Modells [EGP12] mit Meshlab, einer Software für die Verarbeitung von Punktwolken und Oberflächengittern, sowie deren Texturen in einer Vielzahl gängiger Formate.

### 3.3 Algorithmen

In diesem Abschnitt werden das Sparse-Bundle Adjustment zur 3D-Modellbildung und zwei Algorithmen zur Berechnung einer 3D-Oberfläche aus Punkt wolken vorgestellt. Alle drei Algorithmen sind in der RTAB-Map-Anwendung implementiert und wurden im Rahmen dieser Arbeit im Kontext des 3D-Scannens und der Rekonstruktion von 3D-Oberflächen parametrisiert und verwendet.

#### 3.3.1 Sparse-Bundle Adjustment (SBA)

Beim SBA, auch Bündelblockausgleich genannt, werden die Punkte eines Objekts, die durch Aufnahme von sich überdeckenden Bildern dieses Objekts erzeugt wurden, bestimmt. Triggs et al. beschreiben das SBA folgendermaßen: "Das Sparse-Bundle-Adjustment ist das Problem der Verfeinerung einer visuellen Rekonstruktion, um gemeinsam optimale 3D-Struktur- und [...] Be trachtungsparameterschätzungen zu erzeugen. Optimal bedeutet, dass die Parameterschätzun gen durch Minimierung einer Kostenfunktion ge funden werden, die den Modellanpassungsfehler quantifiziert, und gemeinsam mit der Lösung gleichzeitig hinsichtlich der Struktur- als auch der Kameraschwankungen optimal ist. Der Name be zieht sich auf die *Bündel* von Lichtstrahlen, die jedes 3D-Merkmal verlässt und in jedem Kamera zentrum konvergiert, die in Bezug auf Merkmals und Kamerapositionen optimal angepasst wer den. Vergleichbar - im Gegensatz zu unabhän gigen Modellmethoden, die partielle Rekonstruk tionen ohne Aktualisierung ihrer internen Struk tur zusammenführen - werden alle Struktur- und Kameraparameter zusammen in einem Bündel angepasst." [Eigene Übersetzung Tri00, S. 1-2] In Abbildung 3.4 ist die Trajektorie, d.h. der Bewe gungspfad eines Tablets beim 3D-Scannen (türkis), die verschiedenen Kamerablickwin kel als Pyramidenstumpfe (grün) und die aufgenommene Punkt wolke dargestellt.

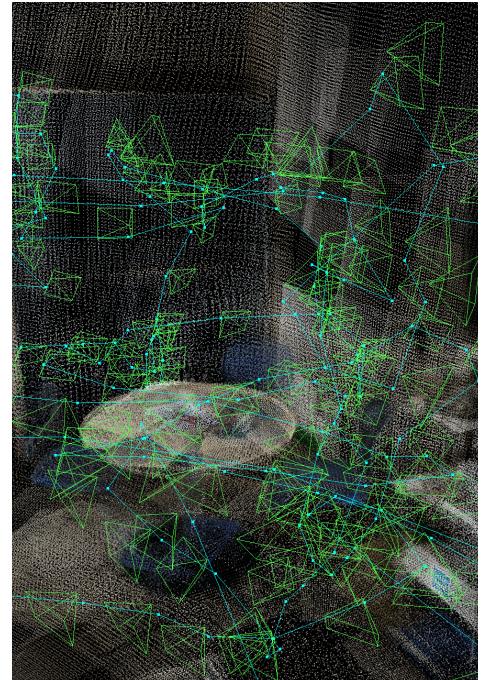
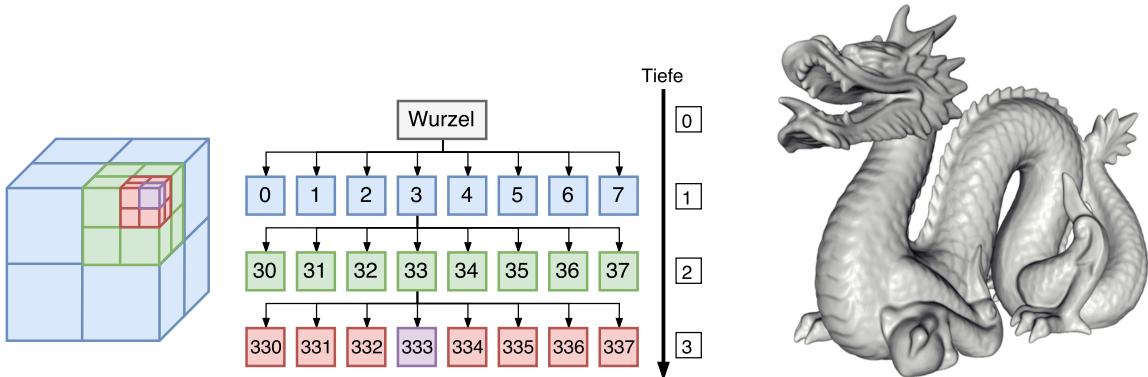


Abbildung 3.4: Darstellung der Trajektorie, d.h. Pfad (blau) und Ausrichtung (grün) der Kamera für das Sparse-Bundle Adjustment (SBA) mit der RTAB-Map-App.

### 3.3.2 Screened Poisson Surface Rekonstruktion (SPSR)

Die Poisson Surface Reconstruction (PSR) schafft wasserdichte Oberflächen aus orientierten Punktwolken [Kaz06]. Wasserdicht bedeutet, dass die Oberfläche keine Löcher aufweist. Dazu wird die Oberflächenrekonstruktion von orientierten Punktwolken als räumliches Poisson-Problem angesehen [Kaz06]. Um das Problem zu lösen, wird der die Punktwolke beinhaltende Quader in alle Richtungen des Raums iterativ in Abhängigkeit zur Dichte der Punktwolke halbiert. So können die Punkte der Punktwolke als die Blätter eines Octrees dargestellt werden [Kaz07]. In Abbildung 3.5a ist ein Octree, der partiell dreimal in alle Raumrichtungen halbiert wurde, dargestellt. Der Octree organisiert die Punktwolke in einer Hierarchie, sodass verschiedene Dichten der Punktwolke durch verschiedene Tiefen des Baumes realisiert werden können. Aus der Topologie des Octree wird eine Menge von binären Bäumen von Kanten abgeleitet. Mit den Kantenbäumen zeigen Kazhdan et al., dass aus dem Octree ein nachweislich wasserndiches Oberflächennetz entnommen werden kann, ohne dass eine Verfeinerung der Knoten oder die Veränderung ihrer Werte erforderlich ist [Kaz07]. Hierfür werden aus den Kantenbäumen Polygone berechnet, welche anschließend trianguliert werden. Dieser Ansatz reduziert die Lösung des Problems auf ein hochdimensionales Gleichungssystem [Kaz06].



(a) Ein Octree repräsentiert als Baum und partiell geteilter Würfel.

(b) 3D-Rekonstruktion eines Drachen mit der SPSR [Bild: Kaz06].

Abbildung 3.5: Ein Octree ist ein Quader, welcher (partiell) in alle Richtungen des Raums iterativ halbiert wird (links). Dabei kann dieses Teilen des Quaders durch einen Baum dargestellt werden, in welchem die Punktwolke hierarchisch organisiert ist. Durch diese Organisation können in Punktwolken Kanten erkannt und damit waserdichte Oberflächen, wie z.B. ein Drache (rechts), berechnet werden.

### 3.3.3 Fast Greedy Projection Triangulation (GP3)

Bei der Greedy-Projection Triangulation (GP3) wird iterativ die kürzeste Kante zwischen zwei beliebigen Punkten, die keine der bestehenden Kanten schneidet, zur initial leeren Menge aller Kanten hinzugefügt. Dies geschieht so lange, bis ein Stopp-Kriterium erreicht wird [Dry95]. Um das Suchproblem zu vereinfachen, werden meist nur Punkte in einem bestimmten Radius für die Suche des nächsten Nachbarn verwendet. Es gibt viele verschiedene Implementierungen dieses Verfahrens, sodass je nach Anwendungsfall eine Implementierung ausgewählt werden kann. Auwahlkriterien sind z.B: die benötigten Ressourcen und Zeit oder ein spezifisches Stopp-Kriterium. In Abbildung 3.6 ist eine Rekonstruktion mit der Greedy-Projection Triangulation dargestellt. Man kann erkennen, dass die Rekonstruktionen der GP3 viele kleine Löcher besitzen, welche das Erscheinungsbild deutlich beeinflussen. Größere Löcher, wie z.B. unter dem Schreibtisch, können vernachlässigt werden, da dort keine Daten während des Scans erfasst wurden.



Abbildung 3.6: Ein mit der GP3 rekonstruierter Arbeitsplatz, der viele kleine Löcher in seiner Oberfläche aufweist. Große Lücken sind zu vernachlässigen, da diese nicht als Punktfolge erfasst wurden.

### 3.4 AR mit Google's Project Tango & Lenovo's Phab 2 Pro

Project Tango ist eine Plattform von Google, die Maschinelles Sehen, auch Computer Vision genannt, verwendet, um Geräten die Fähigkeit zu geben, ihre Position relativ zu der Welt um sie herum zu erfassen. Tango verwendet die Bewegungsverfolgung, das Kartographieren der Umgebung und die Tiefenwahrnehmung der Geräte, um diese im Raum eindeutig zu lokalisieren. Dadurch ermöglicht Tango das Entwickeln von Anwendungen für die erweiterte Realität, auch Augmented Reality (AR) genannt [Goo17c]. Zusätzlich zur Plattform initiierte Google die Entwicklung von Tablets und Smartphones mit integrierten Sensoren für die Tiefenwahrnehmung mit Project Tango. Google beendete Project Tango im August 2017 und ersetzte es durch das Nachfolgeprojekt ARCore [Goo17a]. ARCore setzt auf mit der Handykamera aufgenommene Bilder und entsprechende Bildverarbeitung [Goo17b], um die Orientierung der Geräte im Raum zu bestimmen. Das Phab 2 Pro ist ein Smartphone von Lenovo, das für Googles Project Tango entwickelt wurde. Das Smartphone besitzt eine Tiefenkamera, welche die Distanz zu Gegenständen über die Reflexion von ausgesendetem Licht messen kann. Das Smartphone besitzt ein 6,4 Zoll Display und rückwärtig eine 16 Megapixel RGB-Kamera, einen Tiefenbereichssensor, sowie ein Fisheye-Objektiv. Als Prozessor ist ein für Tango optimierter 64 bit Qualcomm Snapdragon 652 Prozessor mit insgesamt acht CPU-Kernen, davon vier schnelle Cortex-A72-Kerne mit einer Taktfrequenz von 1,8 GHz sowie vier sparsame Cortex-A53-Kerne mit einer 1,2 GHz, eingebaut. Hinzu kommen 4 GB Arbeits- und 64 GB Gerätespeicher [Len17]. Das Tablet ist mit Android als Betriebssystem ausgestattet. Im Rahmen dieser Arbeit wurde das Smartphone zusammen mit der RTAB-Map Android-App für das Verfahren zur Digitalisierung von 3D-Objekten verwendet.

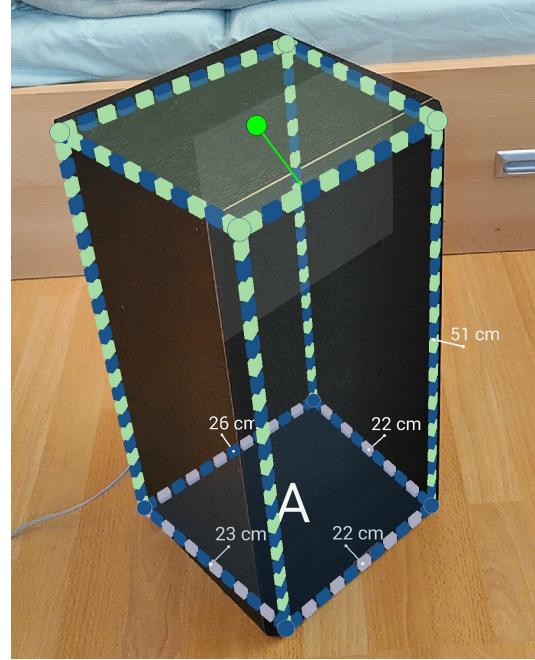


Abbildung 3.7: Lenovo Phab 2 Pro mit einer AR-App aus dem Project Tango zum Vermessen von Gegenständen.

## 3.5 Software zur Entwicklung von VR-Anwendungen

In den nachfolgenden Abschnitten werden die zwei am weitesten verbreiteten Game Engines, Unity 3D (3.5.1) und die Unreal Engine (3.5.2), vorgestellt.

### 3.5.1 Unity 3D

Unity 3D ist ein weitverbreiteter Editor mit integrierter Game Engine, der meist für die Entwicklung von mobilen Anwendungen verwendet wird. „Der Editor ist auf Windows und Mac verfügbar und beinhaltet sowohl eine Reihe von Grafiker-freundlichen Tools für die Erstellung immersiver Erfahrungen und Spielwelten als auch zahlreiche starke Entwickler-Tools für die Implementierung von Spiellogik und für ein erstklassiges Gameplay.“ [Uni17] Unity ist marktführend, was die Unterstützung verschiedenster Plattformen angeht. So erreicht man mit mehr als 25 Plattformen für Mobilgeräte, Desktops, Konsolen, TV, VR, AR und Web das größte Publikum [Uni17]. Ein weiteres Merkmal von Unity ist die Möglichkeit, Werbung in Anwendungen zu integrieren, um Spiele kostenfrei bereitzustellen und durch die dargestellte Werbung Werte zu erwirtschaften, wie es sehr häufig bei den kostenfrei in den gängigen Appstores vertriebenen Spielen praktiziert wird. Zusätzlich bietet es die Möglichkeit, In-App-Käufe für Spiele zu realisieren. Die Game Engine von Unity 3D kann wahlweise durch eine C#- oder JavaScript-API verwendet werden, eine visuelle Skriptsprache gibt es nicht [Uni17]. Die Engine kann kostenfrei getestet werden. Für den professionellen Einsatz müssen Lizenzgebühren in Höhe von 1.380 Euro pro Entwickler und Jahr bezahlt werden.

### 3.5.2 Unreal Engine 4

Die Unreal Engine 4 ist ein Software-Paket aus Entwicklungswerkzeugen, das sich auf hochleistungsfähige Technologie spezialisiert hat. Von mobilen Anwendungen auf Smartphones über Spiele für Konsolen oder kinoähnlichen Visualisierungen bringt die Engine alles mit, was man für die Entwicklung benötigt [Epi1]. Die Engine besitzt eine speziell für VR-Anwendungen entwickelte Rendering-Pipeline, welche die high-quality Berechnung von Beleuchtung und Schattenwürfen sowie hochauflöste Bilder bei 90 Frames per second (FPS) unterstützt [Epi1]. Die Unreal Engine bietet ihre komplette Funktionalität in einer C++-Programmierschnittstelle an. Zusätzlich zur C++-API gibt es eine eigene visuelle Skriptsprache namens Blueprints. Die Blueprints ermöglichen das visuelle Manipulieren der Objekte im Szenengraphen und ihrer Eigenschaften direkt im Unreal Editor [Epi17]. Sowohl die C++-, als auch die Blueprint-API sind umfangreich dokumentiert. Ein eigenes Entwicklerforum ermöglicht den Austausch mit anderen Entwicklern bei Problemen. Die Engine ist kostenfrei erhältlich. Wird ein Produkt mit der Engine entwickelt, so verpflichtet man sich dazu, 5% der Einnahmen an Epic Games abzutreten.

#### 3.5.2.1 Aufbau eines Levels

Eine Szene eines Spiels wird in der Unreal Engine Level genannt. In einem Level werden verschiedene Aktoren positioniert und konfiguriert. Die am häufigsten verwendeten Aktoren sind Static Mesh Actors, Triggers, verschiedene Licht Aktoren, Particle Emitters, Ambient Sound, PlayerStart und Characters. Static Mesh Actors sind Aktoren, die in einem Level platziert werden können und ein Mesh, d.h. eine Oberfläche aus Polygonen, darstellen. Ihr Aussehen kann mit einem Material versehen und in der Engine konfiguriert werden. Die Polygonnetze können in die Engine in den Formaten `obj` und `fbx` importiert werden. Triggers repräsentieren Schalter, die eine Aktion im Level auslösen, jedoch nicht sichtbar sind. Die auszulösende Aktion wird über die Programmierschnittstelle in C++ oder den Blueprints spezifiziert. Es gibt drei verschiedene Arten von Licht, das Point Light, das Spotlight und das Directional Light. Diese entsprechen den in Abschnitt 2.6.3 vorgestellten Lichtern. Weitere Aktoren sind die Particle Emitter, die Partikelsysteme für z.B. Nebel, Rauch und Feuer in einem Level platzieren. Weiterhin gibt es Ambient Sound Aktoren, die das Abspielen einer Audiodatei ermöglichen. Dabei können verschiedene Parameter, wie z.B. Verzerrung, Lautstärke und Geschwindigkeit, angepasst werden. Jedes Level enthält eine sogenannte PlayerStart-Position. Diese repräsentiert die Position, an welcher ein Spieler mit seinem Charakter im Level beim Start erscheint. Mit verschiedenen Standardgeometrien, wie z.B. Quadern, Kugeln, Zylindern und einem Terrain-Generator, können Boden, Wände, Hindernisse und die Umgebung eines Levels erstellt werden.

#### 3.5.2.2 Simulation und Events

Die Engine besitzt für jede Simulation einen festen Ablauf von Events, die ausgelöst werden. Zu Beginn einer Simulation wird genau einmal der sogenannte *BeginPlay*-Event ausgelöst, welcher für die Initialisierung des Levels zuständig ist. Anschließend wird in jedem Simulationsschritt der *Tick*-Event ausgelöst, in welchem das Verhalten der Aktoren spezifiziert ist. Für das Reagieren auf Aktionen oder Situationen während der Simulation können eigene Events definiert werden, die während der Simulation ausgelöst werden, um Verhalten auslagern zu können und gegebenenfalls komplexe Tests der aktuellen Zustände der Aktoren während der Simulationsschritte zu vermeiden. Weitere Events werden durch Eingaben ausgelöst, auf welche sich ein Aktor registrieren kann. Die Engine ruft bei der Auslösung dieses Events alle auf ihn registrierten Funktionen auf. Zusätzlich zu den Events gibt es das sogenannte *Construction*-Script, welches die Aktionen spezifiziert, die ausgeführt werden müssen, um den Aktor zusammenzusetzen, oder seinen Eigenschaften entsprechend zu manipulieren. So kann z.B. bei einem Aktor, der eine Treppe darstellt und den Abstand der Stufen als Eigenschaft besitzt, die Anzahl der Treppenstufen berechnet und diese positioniert werden.

### 3.5.2.3 Die Blueprint-Sprache

In der Unreal Engine können die verschiedenen Aktoren, Materials und Levels durch die eigene Blueprint-Sprache konfiguriert und modifiziert werden. Das Hauptaugenmerk der Sprache liegt auf der visuellen Steuerung des Kontroll- und Datenflusses, der durch bunte Linien dargestellt wird, sowie Knoten, die durch rechteckige Kästen dargestellt werden und entsprechende Ein- und Ausgänge für die Kombination, sogenannte Pins, verfügen. Ein Beispiel für eine Berechnung mit Blueprints ist in Abbildung 3.8 abgebildet. Die Funktion Euklidische Distanz erhält als Eingabe zwei Vektoren namens *Punkt 1* und *Punkt 2*. Ein Vektor besteht in der Unreal Engine aus drei Floats *X*, *Y* und *Z*. In Blueprints können diese komplexeren Datentypen aufgebrochen werden, was durch einen *Break*-Knoten oder direkt im jeweiligen Pin durchgeführt werden kann. In Abbildung 3.8 ist dies für *Punkt 1* direkt am Knoten, für *Punkt 2* mit einem *Break*-Knoten dargestellt. Für die Verarbeitung der Daten werden Knoten aus einer umfangreichen Bibliothek ausgewählt. Für nahezu alle Aktoren gibt es Knoten, um diese spezifisch mit den Blueprints manipulieren zu können. Die Verarbeitung eines Datenflusses erfolgt, wenn er von einem Kontrollfluss als Eingabe benötigt wird. Bei Funktionen, die nur Berechnungen und keine Manipulationen durchführen, erfolgt die Evaluation am *Return*-Knoten.

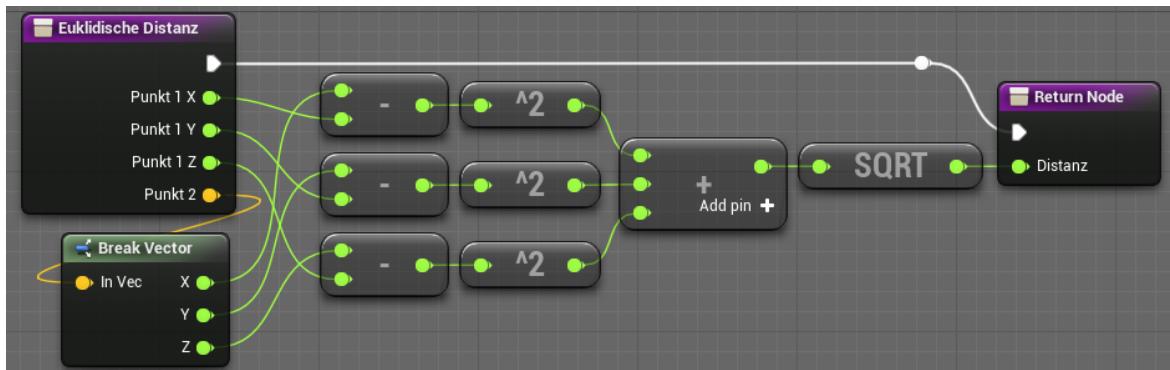


Abbildung 3.8: Die Berechnung der euklidischen Distanz von Blueprints erfolgt schrittweise durch Kombination der Datenflüsse und entsprechende Knoten, die mathematische Operationen durchführen und das Ergebnis an nachgelagerte Knoten weitergeben.

### 3.5.2.4 Erstellen von Aktoren

Wahlweise kann in der Engine ein neuer Aktor als Blueprint oder C++-Klasse angelegt werden. Dabei kann eine Komponente, von der der Aktor abgeleitet wird, angegeben werden. Der Aktor repräsentiert eine Schablone für ein in einem Level platzierbares Objekt und besitzt seinen eigenen Szenegraphen mit Koordinatensystem, damit er aus anderen Aktoren zusammengesetzt werden kann.

## 4 Ergebnisse

Abschnitt 4.1 behandelt die Erkenntnisse, die aus den in Abschnitt 2.2 vorgestellten Forschungsergebnissen hinsichtlich der Information von Patienten mit einer VR-Anwendung gewonnen werden können. Anschließend wird in Abschnitt 4.2 das Verfahren zum Scannen von realen Objekten vorgestellt. Den Abschluss dieses Kapitel bildet die Darlegung der Realisierung des Demonstrators, der die durch das entwickelte Verfahren gescannten Objekte nutzt (Abschnitt 4.5).

### 4.1 Konzept der Information und Aufklärung von Patienten

Für die Aufklärung von Patienten wurde nach den in Abschnitt 2.2 vorgestellten Erfahrungen aus der Praxis ein Konzept für die Aufklärung mittels Virtual Reality erstellt. Dieses gliedert sich in die nachfolgenden Punkte.

1. Digitalisieren und Animieren der in der Strahlentherapie verwendeten Geräte.
2. Simulation von Situationen, in denen der Patient mit diesen Geräten interagiert, wie z.B MRT-Untersuchungen oder eine Bestrahlung.
3. Die Rolle des Patienten in der Simulation soll nahe an seine reale Rolle angelegt sein. Alternativ kann der Patient eine beobachtende Rolle in der Simulation einnehmen. Dabei soll ein virtueller Patient sichtbar sein, sodass er seine eigene Rolle in der Realität erkennen kann.
4. Die Simulation soll sich auf wesentliche Aspekte der zu erklärenden Szene beschränken.
5. Eine Szene soll sich auf ein einziges zu erklärendes Konzept konzentrieren. Komplexere Konzepte sollen durch Kombination mehrerer Szenen erklärt werden, sodass eine Wiederholung einzelner Szenen möglich ist.
6. Der Aufbau einer Szene soll für verschiedene Konzepte wiederverwendet werden, um vom Patienten wiedererkannt werden zu können.
7. Reduzierung von Interaktion und Bewegung des Patienten in der Szene auf ein verständliches Minimum.
8. Einweisung und Erläuterung der Interaktion und des zu erklärenden Konzepts durch einen O-Ton oder das medizinische Personal.

## 4.2 3D-Scannen von Objekten

In diesem Abschnitt werden die Anforderungen und Auswahlentscheidungen des Verfahrens dargestellt. Anschließend werden das Verfahren und seine Komponenten dargelegt.

### 4.2.1 Anforderungen und Auswahl

Für die Digitalisierung eines komplexen Objekts gibt es z.B. das händische Modellieren des Objekts mit einer CAD Software oder das Scannen mit einem 3D-Scanner. Durch das Anwendungsgebiet und die Räumlichkeit ergeben sich Anforderungen an das Verfahren. Die Räumlichkeit ist ein abgeschirmter Bunker eines Linearbeschleunigers. Der Zugang ist durch Strahlenschutzvorschriften stark limitiert und überwacht. Durch Patientenbetrieb, Wartung der Gerätschaften und Forschung ist der Zugang zu den Räumlichkeiten nur in kurzen Blöcken möglich. Weitere Anforderungen ergeben sich durch den Einsatzzweck in einer VR-Anwendung.

1. Der Teil des Verfahrens im Bunker des LINAC muss in begrenzter Zeit durchführbar sein.
2. Durch die Abschirmung der Räumlichkeit muss das Verfahren offline durchführbar sein.
3. Die Farbinformation des Objekts muss erfasst werden.
4. Das 3D-Modell muss in einem verbreiteten 3D-Dateiformat zur Weiterverarbeitung vorliegen.
5. Die Qualität des 3D-Modells muss durch Parametrisierung nachträglich beeinflusst werden können.
6. Das Verfahren soll möglichst kosteneffizient sein und durch Laien durchführbar sein.

All diese Anforderungen können durch verschiedene Ansätze erreicht werden, es wurde jedoch ein Ansatz mit dem Lenovo Phab 2 Pro (siehe Abschnitt 3.4) ausgewählt, da das Tablet kompakt, leicht und mit einem Preis von 500 Euro die preiswerteste Methode ist. Als Software wurde die RTAB-Map Android-App (siehe Abschnitt 3.2.2) ausgewählt.

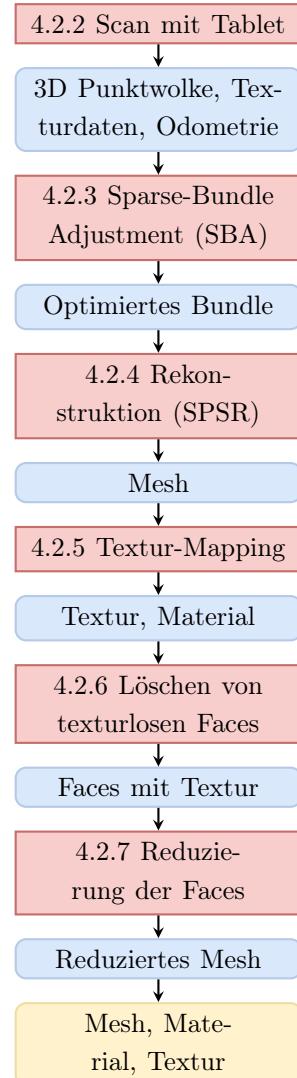


Abbildung 4.1: Der Workflow des 3D-Scannens.

### 4.2.2 Der Scanvorgang

Beim Scannen wird die RTAB-Map App auf dem Tablet gestartet. Nach einer kurzen Initialisierungsphase, in der die App die Orientierung des Tablets durch die Kameras und den Lagesensor des Tablets bestimmt. Zudem wird ein leerer Graph in einem initialen Koordinatensystem angelegt. Die Anzeige des Tablets zeigt dabei ein Grid, das die Orientierung des Bodens aufweist. Anschließend kann durch langsames Bewegen des Tablets schrittweise gescannt werden. Langsam bedeutet dabei, dass das Tablet regelmäßig Fotos mit der Kamera aufnimmt, was länger dauert, als das Bestimmen der entsprechenden Tiefe des Objekts. Zu jedem Kamerafoto wird die Pose und neue Punktfolge mit der aktuellen Tiefenkarte erstellt. Das ist notwendig, damit eine Optimierung des Datensatzes im Nachgang möglich ist. Weiterhin vereinfacht das Einteilen der Punktfolge das Speichern des Datensatzes, da beim Erkennen einer bereits erfassten Struktur die korrespondierende Punktfolge geladen und verglichen angestellt werden können. Dafür muss nicht jederzeit der komplette Datensatz im Hauptspeicher gehalten werden.

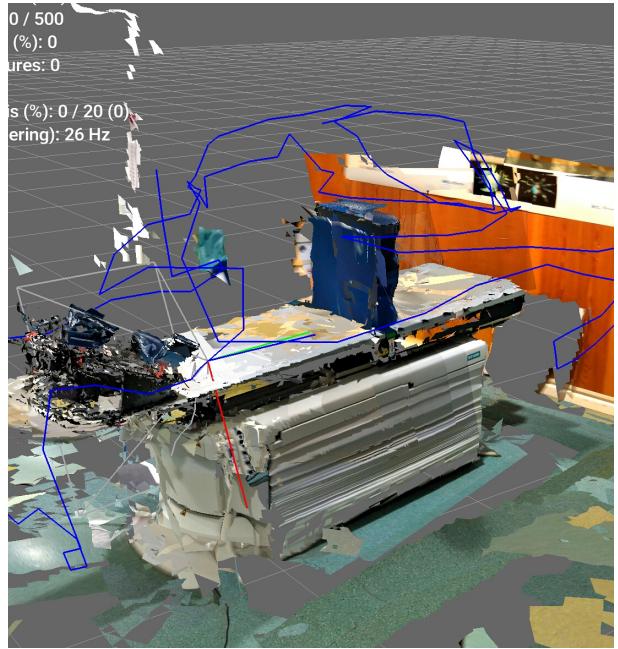


Abbildung 4.2: Ausschnitt der 3D-Visualisierung eines Scans mit der App.

Das Tablet gibt live durch Überblenden der erfassten Umgebung Rückmeldung über die gescannte Bewegung, sodass der Nutzer immer weiß, welche Bereiche des Objekts er bereits erfasst hat (siehe Abbildung 4.2). Wahlweise kann die Punktfolge oder ein live rekonstruiertes, transparentes Mesh angezeigt werden. Das Tablet sollte in verschiedenen Posen und Richtungen um das zu scannende Objekte bewegt werden. Das Scannen kann pausiert werden, um eine andere Position mit dem Tablet einzunehmen zu können. Das wird notwendig, falls andere Gegenstände die Sicht auf das Objekt verdecken sollten und der Nutzer um dieses Objekt herumgehen muss. Würde er den Scan nicht pausieren, würde er den störenden Gegenstand mit erfassen. Das Tablet zeigt dabei das bereits erfasste Mesh an und verändert mit der Trajektorie des Tablets auch die Position und Orientierung des Meshs.

Nachdem das Objekt gescannt wurde, wird der Datensatz auf dem Tablet gespeichert. Anschließend kann eine neue Karte erzeugt und gefüllt werden. Es ist möglich, direkt

mit dem Tablet eine Rekonstruktion zu erstellen oder die Punktwolke zu exportieren. Das Scannen erfolgt mit dem Tablet komplett autonom, d.h. es ist keine Netzwerkverbindung und auch kein verbundener Computer notwendig.

### 4.2.3 Sparse-Bundle Adjustment

Das in Abschnitt 3.3.1 vorgestellte Sparse-Bundle Adjustment (SBA) wird verwendet, um den Datensatz und die Zusammensetzung der einzelnen Punktwolken zu optimieren. Dafür stehen nach einem Scan alle gescannten Punktwolken, die Trajektorie des Tablets und die Kamerafotos zur Verfügung. Das Sparse-Bundle Adjustment optimiert den Graphen und seine Zusammensetzung nicht, da dieser durch den Simultaneous Localization And Mapping (SLAM) Ansatz der Real-Time Appearance-Based Mapping (RTAB-Map)-Anwendung (Abschnitt 3.2.2) und dem darunterliegenden Algorithmus bereits optimal ist. Optimal bedeutet, dass die Punktwolken bereits den Kameraposen zugeordnet wurden und die Karte den Zusammenhang der Punktwolken und Bewegungsdaten durch die Transformationen korrekt abbildet. D.h. die Topologie des Graphen ist bereits korrekt. Das SBA optimiert die zugrundeliegenden Transformationen zwischen den Punktwolken, d.h. die Kanten des Graphen. Optimal heißt hierbei, dass überlappende Punktwolkenbereiche nach dem Adjustment deckungsgleich sind. Das ist notwendig, da Ungenauigkeiten der Sensorik des Smartphones bei der Nachvollziehung der Bewegung nur bedingt während des Scannens korrigierbar sind. Der Optimierer des Sparse-Bundle Adjustment führt 30 Iterationen durch.

### 4.2.4 Rekonstruktion

Für die Rekonstruktion bietet Real-Time Appearance-Based Mapping (RTAB-Map) zwei Algorithmen. Das sind zum einen die Greedy-Projection Triangulation (GP3) und zum anderen die Screened Poisson Surface Reconstruction (SPSR). RTAB-Map bietet auch die Möglichkeit, die Kameraposen mit den Fotos und Punktwolken zu exportieren. Diese können mit z.B. Meshlab eingelesen und verarbeitet werden, falls ein anderer Algorithmus verwendet werden soll. Im Folgenden werden die Ergebnisse der beiden in RTAB-Map enthaltenen Rekonstruktionen an einem Datensatz verglichen und eine Auswahl für das Verfahren getroffen sowie eine mögliche Parametrisierung und ihre Ergebnisse vorgestellt.

#### 4.2.4.1 Das Vergleichsobjekt

Für die Auswahl des Rekonstruktionsalgorithmus wurde ein Scan eines Computertomographen in einem Bestrahlungsraum im DKFZ in Heidelberg erstellt. Zum Vergleich zwischen Scan und realem Objekt ist in Abbildung 4.3 ein Foto (4.3a) des Tomographen neben dem Scan (Abbildung 4.3b) dargestellt. Bei dem Scan wurde nur die

Vorderseite des Tomographen erfasst, da er nicht bewegt wurde. Der Abstand zur Wand zum Tomographen war zu gering, um dessen Rückseite mit dem Tablet erfassen zu können.

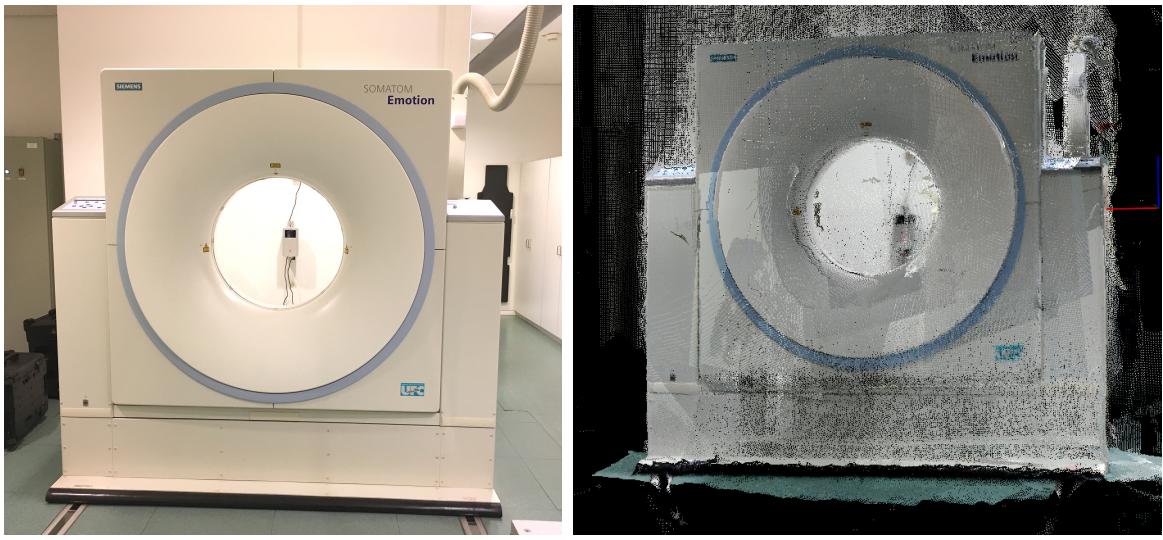


Abbildung 4.3: Foto und kolorierte Punktwolke eines Computertomographen nach der Aufnahme mit der RTAB-Map Android-App und dem Lenovo Phab 2 Pro.

### 4.2.4.2 Parametrisierung der Rekonstruktionen

Für beide Rekonstruktionen werden initiale Vorverarbeitungsschritte durchgeführt. So werden aus den nächsten 24 Nachbarn eines Punkts die Normalen geschätzt. Die Voxelgröße ist auf 1cm eingestellt, was bedeutet, dass die Knoten der Polygone in diesem Radius durch einen einzigen ersetzt werden. Die Punktwolken werden aus dem Datensatz in maximaler Auflösung extrahiert und anschließend um den Dezimierungsfaktor drei reduziert. Dabei wird eine maximale Tiefe von 4m gesetzt, da die Ungenauigkeit des Tiefensors des Smartphones ab dieser Tiefe zunimmt (experimentell bestimmt). Das Tiefenbild wird mit einem Gaußfilter gefiltert (Größe 10, SD=0,1). Die Parametrisierung der Greedy-Projection Triangulation (GP3) erfolgt so, dass für die Suche des nächsten Nachbarn ein Kugelradius von 8,75cm verwendet wird. Dies begrenzt den Suchraum und ist möglich, da die Punktwolke eine ausreichende Dichte besitzt. Die Parametrisierung der Screened Poisson Surface Reconstruction (SPSR) erfolgte so, dass eine maximale Tiefe des Octree von 9 festgelegt wurde. Die minimale Tiefe betrug 6. Für den nachfolgenden Vergleich wird eine Rekonstruktion mit einer minimalen Tiefe von 9 durchgeführt, um den Einfluss dieses Parameters auf das Ergebnis der Screened Poisson Surface Reconstruction zu verdeutlichen. Durch die hohe Dichte

der Punktfolge wird festgelegt, dass sich mindestens 15 Punkte in einem Knoten des Octree sammeln müssen.

### 4.2.4.3 Ergebnisse der Rekonstruktionen

Das Ergebnis der Rekonstruktion mit Greedy-Projection Triangulation (GP3) ist in Abbildung 4.4a dargestellt, das der Screened Poisson Surface Reconstruction (SPSR) in Abbildung 4.4b und 4.4c. Weiterhin findet sich in Tabelle 4.1 eine Auflistung der für die Rekonstruktion benötigten Zeit, Anzahl der Vertices und Faces sowie die finale Dateigröße des rekonstruierten 3D-Modells. Der Originaldatensatz besitzt eine Größe von 167MB.

Aus Tabelle 4.1 kann entnommen werden, dass die Rekonstruktion durch die Greedy-Projection Triangulation (GP3) deutlich länger dauert, als die Rekonstruktion mit der Screened Poisson Surface Reconstruction (SPSR) bei einer minimalen Octree-Tiefe von 6. Erhöht man die Tiefe des Octrees der SPSR, so steigt die für die Rekonstruktion benötigte Zeit an. Die für die GP3 benötigte Zeit steigt bei Veränderung ihrer Parameter ebenfalls an, jedoch kann man nicht direkt steuern, ob das Ergebnis der Rekonstruktion genauer wird. Aus der Tabelle kann ebenfalls entnommen werden, dass die Rekonstruktion der GP3 deutlich mehr Eckpunkte und Faces generiert und deshalb auch die Dateigröße dieser Rekonstruktion größer als die der SPSR bei vergleichbarem Detailgrad 6 ist. Betrachtet man die Ergebnisse der Rekonstruktionen (4.4a, 4.4b), so wird ersichtlich, dass die SPSR größere Dreiecke als die GP3 produziert. Dadurch, dass sie wasserdichte Oberflächen erzeugt, besitzt sie jedoch eine abgeschlossene Oberfläche. Die GP3 produziert viel feinere Dreiecke, besitzt aber deutlich mehr Löcher in ihrer Oberfläche. Erhöht man die minimale Tiefe des Octrees der SPSR auf 9, so erhöhen sich der Ressourcenbedarf sowie die benötigte Zeit für die Rekonstruktion exponentiell. Die Oberfläche der Rekonstruktion in Abbildung 4.4c bleibt direkt nach der Rekonstruktion wasserdicht, durch das Entfernen der texturlosen Faces entstehen im Bereich des Sockels Löcher, jedoch weitaus weniger als bei der Greedy-Projection Triangulation. Es gibt texturlose Faces vermehrt an Kanten. Dort kann es vorkommen, dass kein Kamerafoto einem Teilbereich einer Punktfolge aufgrund zu spitzer Winkel zugeordnet werden kann. In der Regel werden solche Fälle durch überlappende Kamerabilder abgefangen, die an den Kanten jedoch nicht zwingend vorhanden sein müssen. Aus Tabelle 4.1 kann entnommen werden, dass bei der Rekonstruktion in Abbildung 4.4c mehr als dreimal so viele Vertices, jedoch nur 20% mehr Faces im finalen Datensatz enthalten sind, als z.B. bei der Greedy-Projection Triangulation in Abbildung 4.4a. Dieses Phänomen kann im Vergleich zur Rekonstruktion mit der Tiefe 6 in Abbildung 4.4b noch deutlicher quantifiziert werden, da ca. 19,6 mal so viele Vertices, aber nur fünf mal so viele Faces im Modell vorhanden sind. In der Abbildung selbst kann dies im Bereich der Öffnung des Computertomographen beobachtet werden, ein deutlicheres Rauschen, als in Abbildung 4.4b ersichtlich ist.



(a) Greedy-Projection Triangulation (b) SPSR (min 6) (c) SPSR (min 9)

Abbildung 4.4: Ergebnisse der Rekonstruktion des Computertomographen

| Modalität      | Dauer   | Anzahl Eckpunkte | Anzahl Flächen | Dateigröße |
|----------------|---------|------------------|----------------|------------|
| Original       | -       | -                | -              | 166.937kB  |
| Fast-GP3       | 0:01:15 | 236.558          | 294.357        | 48.320kB   |
| SPSR-9 (min 6) | 0:00:17 | 45.984           | 69.698         | 10.281kB   |
| SPSR-9 (min 9) | 1:56:08 | 902.669          | 355.091        | 81.664kB   |

Tabelle 4.1: Vergleich der Rekonstruktionen in Bezug auf die für die Rekonstruktion benötigte Zeit und Dateigröße.

#### 4.2.5 Textur-Mapping

Nach der Rekonstruktion wird ein Textur-Mapping durchgeführt. Dabei werden die Kameraposen auf die Faces der Punktewolken projiziert und überlappende Kamerafotos nach einer Filterung des Kamerawinkels zusammengeführt und perspektivisch entzerrt. Die maximale Distanz der Polygone zur Kamera beträgt 4m. Die Helligkeit der Kamerabilder wird um maximal 10% automatisch angeglichen. Das Ergebnis dieses Schritts ist die Zuordnung der Texturkoordinaten zu den Faces sowie Informationen zum Material der Oberfläche, welche nach diesem Verarbeitungsschritt immer dieselben sind, da keine Spiegelungen oder speziellen Transformationen durchgeführt werden sowie keine Materialeigenschaften, wie z.B. Reflexion oder Transparenz, gesetzt werden. Beim Speichern der Textur werden alle Kamerabilder, von denen Ausschnitte zur Texturierung ausgewählt sind, in einer großen Bilddatei angeordnet. Ein Beispiel für solch eine Textur ist in Abbildung 4.5 dargestellt. Mit der Software ist es möglich, bis zu zwanzig Texturen der Auflösung 32768x32768 Pixel<sup>2</sup> zu berechnen. Die Größe der Textur wird dadurch limitiert, dass die Unreal Engine Texturen mit einer maximalen Auflösung von 8192x8192 Pixel<sup>2</sup> unterstützt. Eine Textur mit dieser Auflösung besitzt eine Größe von circa 90MB.

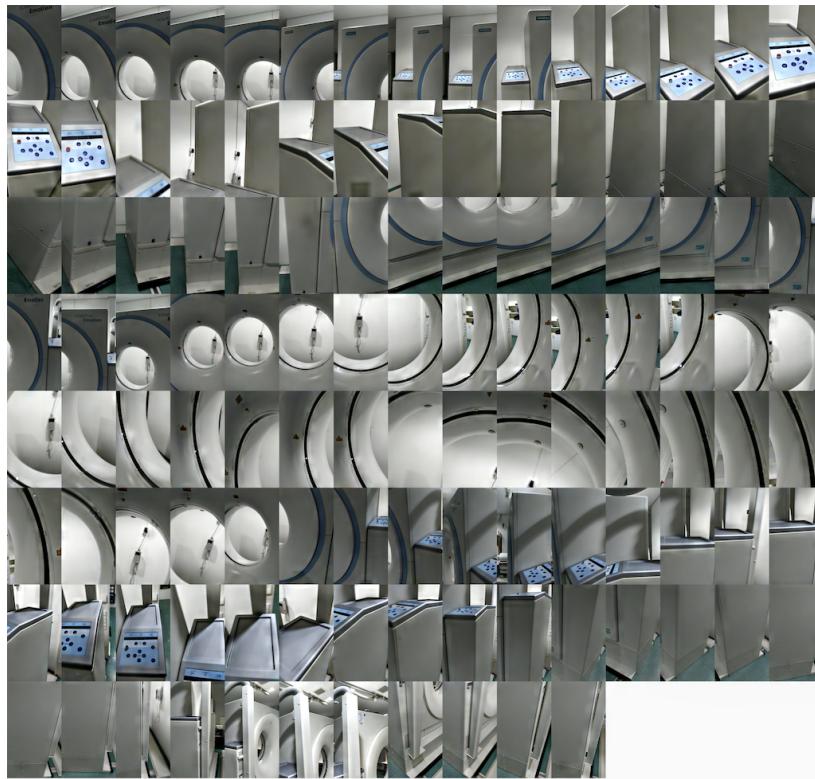


Abbildung 4.5: Die Textur ist aus einzelnen Kamerabildern zusammengesetzt.

Ein Ausschnitt einer solchen Textur kann in Abbildung 4.6 betrachtet werden. Das Bild wurde erzeugt, indem in der virtuellen Welt aus einer Entfernung von etwa 5 virtuellen Metern ein High-Resolution Screenshot erzeugt wurde. Der Screenshot selbst besitzt eine Auflösung von 4320x2400 Pixel<sup>2</sup>. In der Abbildung sind Stellen, an denen die Texturen zueinander verschoben zusammengesetzt sind mit orangefarbenen Pfeilen markiert. Diese Verschiebungen entstehen bei der Auswahl der Kamerafotos in deren überlappenden Bereichen. Stellen mit guter Auflösung sind mit grünen Pfeilen versehen, Bereiche mit unterschiedlicher Helligkeit mit lilafarbenen Pfeilen.



Abbildung 4.6: Darstellung eines Texturausschnitts in einem Virtual Environment.

#### 4.2.6 Löschen von Faces ohne Textur

Beim Löschen von Faces ohne Textur werden die Teile der Modelle entfernt, die keine Textur besitzen, d.h. ohne Kamerabild rekonstruiert werden. Das passiert, da erst nach der Rekonstruktion die Kameraposen evaluiert und die Textur erstellt werden. Werden alle Kameraposen bei der Texturierung an Kanten auf Grund eines zu spitzen Winkels zur Oberfläche ausgeschlossen, werden diese weiß dargestellt. Gleichzeitig neigt die SPSR dazu, eine Hülle um alle gescannten Objekte zu rekonstruieren, um die Eigenschaft der Abgeschlossenheit erfüllen zu können. Diese Eigenschaft muss erhalten bleiben, da auf ihr die Annahmen für die Vereinfachung der Rekonstruktion basieren und erst dadurch die Linearisierung des hochdimensionalen Gleichungssystems möglich ist. Ein Beispiel dafür ist in Abbildung 4.7 dargestellt. In der Abbildung

erkennt man deutlich die erzeugte Hülle um den Computertomographen. Zusätzlich sind weitere an sich abgeschlossene Objekte erkennbar, welche auf dichtere Bereiche in der gescannten Punktfolge zurückzuschließen lassen. Diese Artefakte werden nicht texturiert und können so in diesem Schritt automatisch entfernt werden. Diese bedeutet, dass an allen Objekten alle nicht texturierten Faces entfernt werden, sodass diese unter Umständen nicht mehr abgeschlossen sind und Löcher erzeugt werden. Ist dies nicht erwünscht, so kann man das Modell auch nachträglich mit Meshlab (3.2.3) ausschneiden und nicht erwünschte Faces löschen.

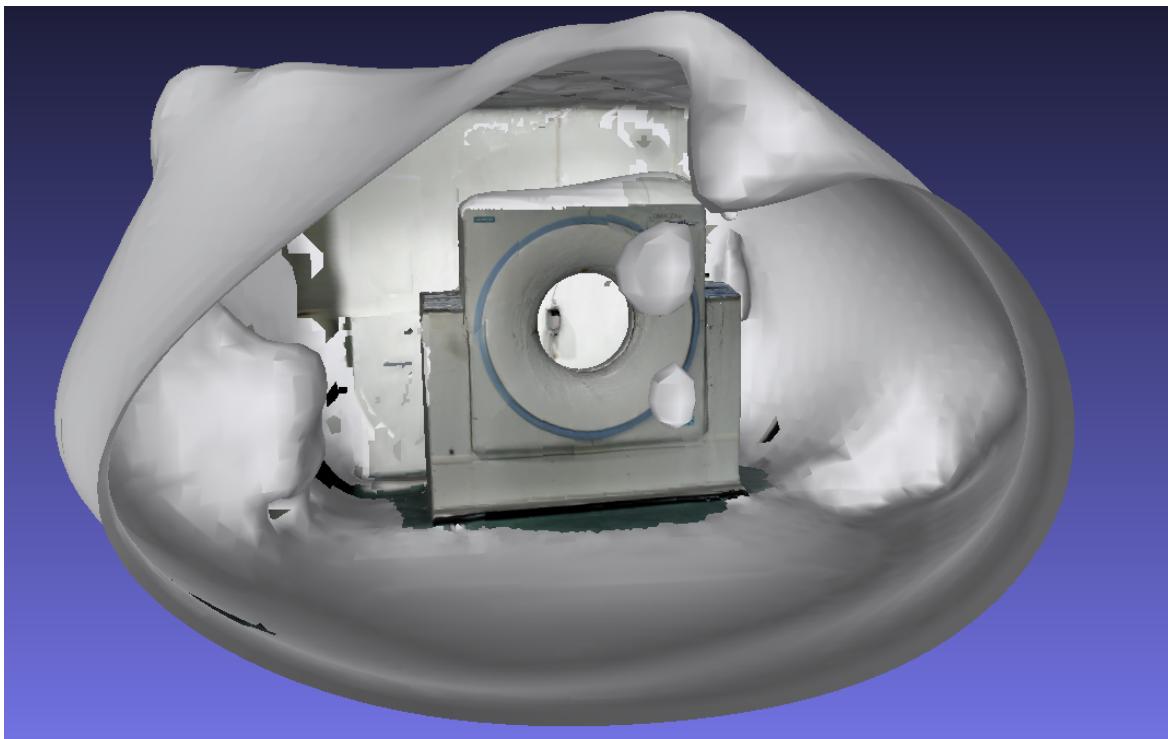


Abbildung 4.7: Darstellung einer Rekonstruktion inklusive untexturierter Faces.

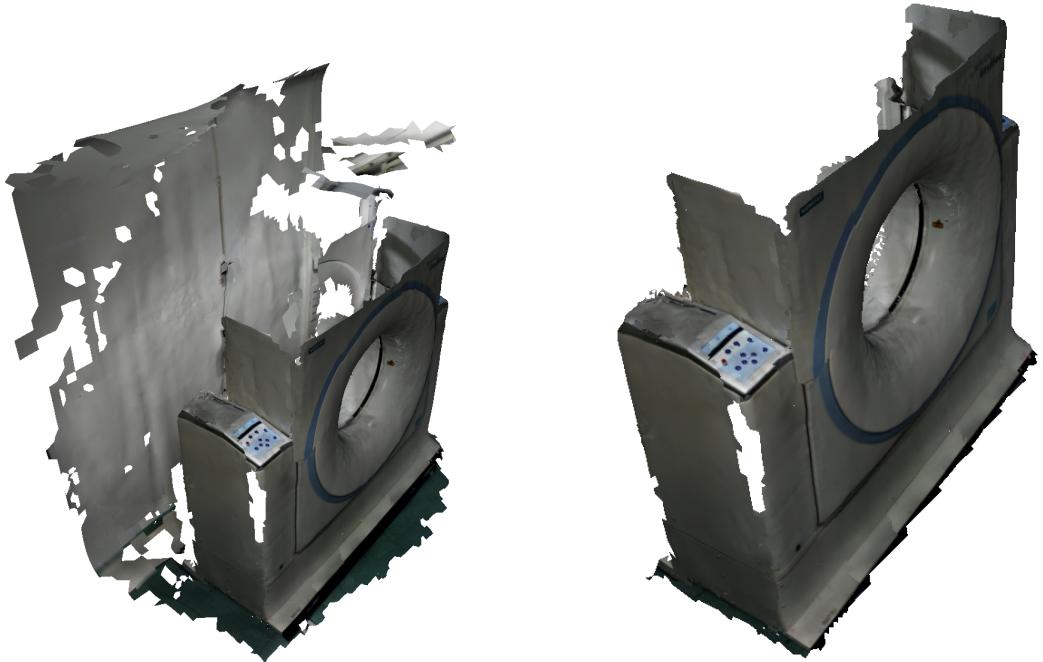
### 4.2.7 Reduzierung der Faces

In einem experimentellen Versuch mit rekonstruierten Meshes auf der gegebenen Hardware wurde herausgefunden, dass in einer Szene bis zu 2.5 Millionen texturierte Faces dargestellt werden können, ohne diese Schwelle von 90 Frames per second (FPS) für eine flüssige Darstellung der Szene zu unterschreiten. Diese Schwelle ist abhängig von der Leistungsfähigkeit des verwendeten Computer, sollte jedoch als Richtwert betrachtet werden. Die Rekonstruktion kann bei einer minimalen Baumtiefe von 6 oder größer ein Modell mit z.B. 1.8 Millionen Faces und einer Dateigröße von 1.2GB (ohne Textur) erstellen. Wenn man davon ausgeht, dass in der Szene häufig mehrere Objekte dargestellt werden, ist die Zahl der maximalen Faces pro Modell auf 200.000 festgesetzt. Bei der Reduzierung der Faces werden diese schrittweise zusammengeführt. Dieser

Schritt erfolgt nach der Texturierung, sodass das Entstehen von verschobenen Kanten verschiedener Kamerabilder nicht berücksichtigt werden kann.

### 4.2.8 Veredelung des 3D-Modells

Beim Scannen der Objekte wird nicht zwischen Hintergrund, Bewegungsartefakten und dem gewünschten Objekt unterschieden, weshalb eine Veredelung des rekonstruierten Datensatzes notwendig ist. Hierbei werden Oberflächen entfernt, die nicht sichtbar sind, da sie durch andere Oberflächen verdeckt sind. Weiterhin werden die Objekte unten beschnitten, sodass der verbleibende Anteil des Objekts in der x-y-Ebene des Koordinatensystems eben aufliegt (siehe Abbildung 4.8). Kleine Objekte, die durch den Scan nur schlecht erfasst sind, werden ebenfalls entfernt. Dadurch kann es zu Löchern in der verbleibenden Oberfläche kommen. Weiterhin werden Objekte aus Scans zusammengefasst oder Objekte, die nur durch mehrere Scans erfassbar waren, zusammengesetzt. Dadurch, dass RTAB-Map nach einer einmaligen Initialisierung für die nachfolgenden Scans denselben Koordinatenursprung wählt, können die Objekte zusammen importiert werden, um relativ zueinander richtig positioniert zu sein.



(a) Rekonstruktion mit Artefakten

(b) Manuell gesäuberte Rekonstruktion

Abbildung 4.8: Das Ergebnis des manuellen Veredelns eines gescannten Datensatzes mit Meshlab ist ein Modell ohne Artefakte und mit bereinigten Kanten. Anschließend kann es direkt in ein Virtual Environment importiert werden.

### 4.3 Aufbau eines HTC Vive VR-Systems

Folgt man der HTC Aufbauanleitung [Val16], so besteht die minimale Ausstattung eines VR-Systems aus einem leistungsstarken Gaming-Computer mit einer modernen Grafikkarte sowie der VR-Hardware. Dazu werden ein Monitor und gewöhnliches Zubehör, wie z.B. Tastatur und Maus, benötigt. Die VR-Hardware besteht aus zwei Tracking-Kameras, sogenannten Base Stations, und dem Head-Mounted Display (HMD) sowie zwei Controllern und verschiedenem Zubehör. Die beiden Base Stations können mit zwei verschiedenen Synchronisationsmodi verwendet werden. In einem optischen Modus ist eine Sichtverbindung der beiden Base Stations notwendig. In einem kabelbasierten Modus ist eine Kabelverbindung der beiden Basisstationen erforderlich. Das VR-System kann in zwei verschiedenen Raumkonfigurationen verwendet werden. Steht viel Platz (bis zu 5x5m) zur Verfügung, so kann das System in diesem Raum die ganze Fläche verwenden, sodass man sich in der virtuellen Welt, begrenzt auf diesem Bereich, bewegen kann. Steht wenig Platz zur Verfügung, so kann das System im Stand mit kleinem Bewegungsradius verwendet werden, sodass ein Umdrehen aber kein Laufen möglich ist. Für einen portables System kommt man zu dem in Abbildung 4.9 dargestellten Aufbau. Es wird ein Platz von 3x3m für Tracking des HMD und der Controller benötigt. Dieser Raum ist in einen kreisförmigen Bereich mit einem Durchmesser von etwa 2m unterteilt. Dies garantiert, dass die Basisstationen immer einen Mindestabstand von 1m besitzen, um den Abstand zum HMD und den Controllern bestimmen zu können. Die Base Stations werden in einer Höhe von etwa 2m aufgestellt. Es wurde das kabelbasierte Tracking ausgewählt, da durch die kleine Distanz und die geringe Höhe der Base Stations häufig keine Sichtverbindung zwischen diesen möglich ist. Dies resultiert daraus, dass die Kameras nach unten gekippt werden müssen, um den Trackingraum überblicken zu können und dann nicht frontal aufeinander ausgerichtet sind.

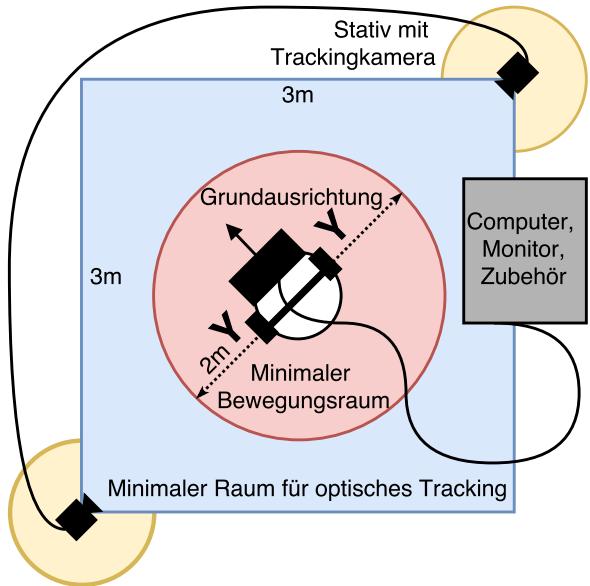


Abbildung 4.9: Der minimale Aufbau des VR-Systems besteht aus einer freien Fläche von  $3 \times 3\text{m}^2$  für das optische Tracking. Darin gibt es einen Bewegungsraum in Form eines Kreises mit 2m Durchmesser.

Es wird ein Platz von 3x3m für Tracking des HMD und der Controller benötigt. Dieser Raum ist in einen kreisförmigen Bereich mit einem Durchmesser von etwa 2m unterteilt. Dies garantiert, dass die Basisstationen immer einen Mindestabstand von 1m besitzen, um den Abstand zum HMD und den Controllern bestimmen zu können. Die Base Stations werden in einer Höhe von etwa 2m aufgestellt. Es wurde das kabelbasierte Tracking ausgewählt, da durch die kleine Distanz und die geringe Höhe der Base Stations häufig keine Sichtverbindung zwischen diesen möglich ist. Dies resultiert daraus, dass die Kameras nach unten gekippt werden müssen, um den Trackingraum überblicken zu können und dann nicht frontal aufeinander ausgerichtet sind.

## 4.4 Auswahl einer Game Engine

Die Anforderungen für die Game Engine ergeben sich zum einen durch die vorgegebene Hardware und zum anderen durch die Leistungsanforderungen der zu entwickelnden VR-Anwendung. Als Hardware ist ein Desktop-Computer mit Windows 10 als Betriebssystem sowie ein HTC Vive bzw. SteamVR HMD, im Weiteren als HTC Vive benannt, vorgegeben. Der Desktop-Computer besitzt eine Intel Core i7-6700K Central Processing Unit (CPU), die mit vier Kernen bei jeweils 4.0 GHz getaktet ist, 64 GB Arbeitsspeicher sowie eine Nvidia GTX 1080 Grafikkarte mit 4 GB Grafikspeicher.

1. Die Game Engine muss die VR-Hardware mit voller Funktionalität unterstützen.
2. Zusätzlich zur Engine muss es ausreichend Dokumentation zur Programmierschnittstelle geben, sodass auch unerfahrene Benutzer damit umgehen können.
3. Die Engine und ihre Dokumentation sollen vollständig kostenfrei verfügbar sein.
4. Die Engine soll möglichst quelloffen sein.
5. Die Engine soll möglichst verbreitet sein, um eine Weiterentwicklung und kontinuierliche Beseitigung von Fehlern sicherzustellen.
6. Die Engine soll eine API in der Programmiersprache C++ bereitstellen.

Für die Auswahl der Game Engines werden die CryEngine, Unity 3D und die Unreal Engine berücksichtigt, da diese die am weitesten verbreiteten Game Engines auf dem Markt sind. Die CryEngine konnte ausgeschlossen werden, da diese noch keine vollständige Unterstützung von VR gewährleistet [Cry16]. Die Auswahl fiel auf die Unreal Engine 4, da diese eine Programmierschnittstelle in C++ bereitstellt und eine für Virtual Reality optimierte Rendering-Pipeline besitzt, sodass die Leistung des Systems besser ausgenutzt werden kann, als dies mit Unity möglich wäre. Das bedeutet konkret, dass durch diese Pipeline und eine damit optimierte Verarbeitung der 3D-Modelle mehr Faces in einer Szene platziert werden können, ohne die Schwelle von 90 Frames per second zu unterschreiten. Zudem ist die Dokumentation der Unreal Engine mit zusätzlichen Tutorials und Videos umfangreicher als die von Unity 3D. Ein weiterer Aspekt der Auswahl der Unreal Engine ist, dass auch ohne Programmierung mit C++ die Entwicklung von Simulationen möglich ist. Dies kann von Vorteil sein, damit nicht zwingend Informatiker, sondern auch das medizinische Personal Simulationen erstellen kann. Die Blueprint-Sprache und das visuelle Skriptingsystem direkt im Unreal Editor reduzieren die Einarbeitungszeit und Fehleranfälligkeit der Software.

## 4.5 Entwicklung des Demonstrators

Ein Ziel dieser Arbeit war es, einen lauffähigen Demonstrator zu entwickeln, der die durch das entwickelte Verfahren digitalisierten 3D-Modelle enthält. Das Konzept des Demonstrators besteht darin, eine Szene zu erstellen, in welcher das Zusammenwirken mehrerer Objekte demonstriert wird. Die Szene soll modellierte und gescannte Objekte enthalten und an einen realen, in der Strahlentherapie angelehnten, Einsatz erinnern, um realistisch zu demonstrieren, dass das Verfahren anwendbar ist. Dafür wurde ein Bestrahlungsraum im Deutschen Krebsforschungszentrum in Heidelberg ausgewählt. Der Bestrahlungsraum beinhaltet einen Elektronenlinearbeschleuniger mit Patientenliege sowie einen zusätzlich vorhandenen Computertomographen. Das Ergebnis der entwickelten Szene ist in Abbildung 4.10 dargestellt. Weiterhin sollen durch die Entwicklung des Demonstrators die Grenzen und Limitationen der Game Engine im Kontext von Virtual Reality evaluiert werden.

Nachfolgend wird demonstriert, wie wie komplexe reale Abläufe und Bewegungen mehrerer Geräte visuell und akustisch simuliert werden können. Dabei wird auf die Möglichkeiten der Realisierung dieser zeitlich abhängigen Simulation mit der Game Engine eingegangen. Dazu wird in Abschnitt 4.5.1 zunächst auf die Funktionalität des Demonstrators eingegangen und dargelegt, welche Aspekte der Realität abgebildet wurden. Im nachfolgenden Abschnitt werden daraufhin die Akteure, die diese Funktionalität bereitstellen, eingegangen. Anschließend wird das Datenmodell, welches für die Parametrisierung der komplexen Bewegungsabläufe verwendet wird, vorgestellt. Im Anschluss werden die einzelnen Funktionen der Akteure und deren Umsetzung mit der Game Engine erläutert.

### 4.5.1 Funktionalität des Demonstrators

Die Funktionalität des Demonstrators besteht aus der Drehung der Gantry des Elektronenlinearbeschleunigers sowie der Bewegung der Patientenliege. Die Gantry kann sich dabei um einen angegebenen Winkel entlang ihrer Drehachse drehen. Der Bestrahlungsstrahl kann wahlweise ein- oder ausgeschaltet werden. Der Status des Elektronenlinearbeschleunigers wird an einer Anzeige dargestellt.

Es wird zwischen *Ruhephase*, *Bewegung der Gantry* und *Strahl an* mit aufsteigender Priorität unterschieden. Der Elektronenlinearbeschleuniger besitzt zwei Positionierungslaser, deren Strahl orthogonal zur Bewegungssachse des Beschleunigers durch das Isozentrum zeigt. Die Laser sowie die Statusanzeige des Beschleunigers werden bei der Platzierung des Beschleunigers in einer Szene automatisch platziert, sodass diese immer an einem Objekt befestigt sind und nicht in der Luft schweben. Die Patientenliege besteht aus 5 Segmenten, wovon zwei beweglich sind. Eines der unteren Segmente



Abbildung 4.10: Vogelperspektive der Szene des Demonstrators mit modelliertem Linearbeschleuniger, Patientenliege sowie mehreren gescannten Objekten, wie z.B. dem Computertomographen. ↗

ist in vertikaler Richtung beweglich, die Liege in horizontaler Richtung. Die Patientenliege wird mit dem Linearbeschleuniger automatisch in der Szene positioniert, sodass ihre relative Position und Höhe als Ausgangslage zum Beschleuniger immer die selbe ist. Patientenliege und Elektronenlinearbeschleuniger sind komplett unabhängig voneinander bewegbar.

Beide, Elektronenlinearbeschleuniger und Patientenliege, sind nicht nur durch ihre Oberflächen und Bewegungen in der Welt wahrnehmbar, sondern besitzen individuelle Geräusche. Der Elektronenlinearbeschleuniger besitzt dabei ein immer vorhandenes Hintergrundgeräusch, das in der Realität durch seine Kühlung verursacht wird. Bewegt sich die Gantry, ist ein Geräusch eines Elektromotors hörbar. Ist der Strahl des Elektronenlinearbeschleunigers an, so hört man ebenfalls ein typisches Geräusch. Bei der Bewegung der Patientenliege kann beim Start der Bewegung ein Klicken und anschließend ein Bewegungsgeräusch wahrgenommen werden. Dieses ist bis zum Ende der Bewegung vorhanden und wird durch ein erneutes Klicken beendet, um das reale

Ein- und Ausschalten eines Motors nachzuahmen. Die Bewegungsgeräusche sind in vertikaler und horizontaler Richtung unabhängig zueinander und lauter, falls beide Bewegungen gleichzeitig durchgeführt werden. Die Geräusche von Patientenliege und Elektronenlinearbeschleuniger sind an die Objekte und deren Position im Raum gebunden. Das bedeutet, dass diese bei einer Bewegung des Nutzers im Raum entsprechend an Lautstärke zu- und abnehmen. Zusätzlich zu den im Raum vorhandenen Objekten und Geräuschen gibt es eine Hintergrundstimme, die mit konstanter Lautstärke allgemeine Dinge über den Demonstrator und seinen Zweck erzählt und Verhaltensregeln für die VR-Nutzer bereitstellt.

Nutzer können durch Kopfbewegungen ihre Orientierung in der Szene verändern. Da ein freies Bewegen in der Szene durch räumliche Einschränkungen der realen Welt und die Kabelverbindung des Head-Mounted Displays nicht möglich ist, gibt es eine Teleport-Funktion, um andere Positionen im Raum einzunehmen. Diese Funktion wurde von einer Beispielanwendung der Unreal Engine übernommen und leicht angepasst und wird im Folgenden nicht behandelt.

Die beschriebene Funktionalität kann nur erreicht werden, wenn die verschiedenen Objekte und Geräusche in einer zeitlichen Abfolge zueinander agieren. Dafür ist die Definition eines Datenmodells notwendig. Für die Parametrisierung des Datenmodells werden häufig anfallende Parameterbelegungen, die nur noch wenige Freiheitsgrade besitzen, vereinfacht. Ein Beispiel dafür ist die Bestrahlung. Angenommen die Gantry des Linearbeschleunigers und der Tisch bewegen sich dabei nicht, muss die Zeit der Bestrahlung parametrisiert und der Strahl angeschaltet werden.

### 4.5.2 Aktoren der Szene

Die in Abschnitt 4.5.1 beschriebene Funktionalität wird von fünf Aktoren bereitgestellt, welche in Abbildung 4.11 abgebildet sind. Diese werden gemeinsam in der Szene positioniert. Der Linearbeschleuniger mit seinem feststehenden Sockel und der drehbaren Gantry steuert dabei die Statusanzeige, die Positionierungslaser und den Bestrahlungsstrahl. Die Patientenliege besteht aus insgesamt fünf Segmenten und einem festen Sockel. Die Segmente sind von unten nach oben hierarchisch im Szenographen organisiert, sodass eine Transformation eines unteren Segments auf alle darüber liegenden Segmente ebenfalls angewandt wird. Das Segment 2A, welches in Abbildung 4.11 rot markiert ist, kann in vertikaler Richtung bewegt werden. Das 3D-Modell ist dupliziert eingefügt, damit der Tisch bei der Bewegung nicht beginnt zu schweben. Segment 5, welches in Abbildung 4.11 blau markiert ist, repräsentiert die aufliegende Platte der Liege und kann in vertikaler Richtung zum Linearbeschleuniger bewegt werden.

## 4 Ergebnisse

---

Die Integration von digitalisierten Objekten erfolgt mit der Unreal Engine durch Import einer `obj`-Datei der veredelten Modelle. Die Engine importiert das Material und die Textur des Objekts automatisch. Beim Import muss die Transformation des Modells, insbesondere Skalierung und Rotation, angepasst werden, da ein Meter eines Scans auf einen Zentimeter in der Engine abgebildet wird. Durch den entsprechenden Skalierungsfaktor beim Import kann dieses Problem behoben werden. Zudem sind im Unreal Koordinatensystem zwei Achsen vertauscht. Nach dem Import muss die Qualität der gerenderten Textur für die Simulation und das Kollisionsmesh angepasst werden, da dieses standardmäßig ein Quader ist. Dies verhinderte zunächst, dass der Positionierungslaser durch die Öffnung des Computertomographen strahlte, da eine Kollision kurz vor dem Tomographen erkannt wurde. Jedes importierte Mesh wird in der Engine als *Static Mesh* repräsentiert, sodass diese äquivalent wie intern erzeugte Meshes in den Blueprints oder der C++-API behandelt werden können. Diese Schritte nehmen pro Modell etwa zehn Minuten in Anspruch, sodass der Zusatzaufwand vernachlässigt werden kann.

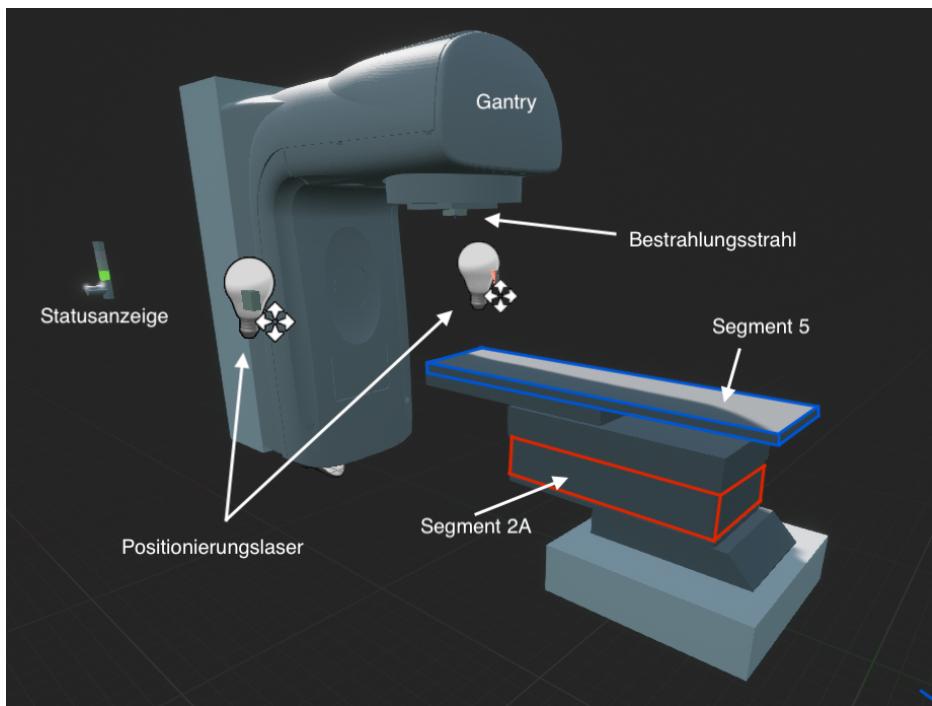


Abbildung 4.11: In der Abbildung ist der Aufbau der funktionalen Akteure der 3D-Szene abgebildet. Der Linearbeschleuniger steuert seine drehbare Gantry, die Statusanzeige, den Bestrahlungsstrahl sowie die Positionierungslaser. Die Patientenliege besteht aus mehreren Segmenten, von denen zwei bewegt werden können.

#### 4.5.3 Datenmodell für Animation und Interaktion

Das Datenmodell des Demonstrators ist in Abbildung 4.12 abgebildet. Da eine 3D-Szene einer Anwendung zur Information und Aufklärung von Patienten oder zur Lehre von angehenden Ärzten, MTA und Studierenden immer einen Inhalt vermitteln soll, besitzt eine Szene ein Skript. Ein Skript kann man sich ähnlich wie ein Drehbuch in der Filmindustrie vorstellen. Das Skript besitzt verschiedene filmische Szenen und beschreibt, was in welcher zeitlichen Abfolge wo geschehen soll. In der Anwendung wird diese Abfolge durch Bewegungskonfigurationen (*MovementConfiguration*) realisiert. Eine solche Bewegungskonfiguration besitzt dabei die Konfigurationen aller Objekte der Szene, die ihre Zustände verändern können. Im Demonstrator sind das der Linearbeschleuniger mit den beiden Positionierungslasern, die Statusanzeige und der Bestrahlungsstrahl sowie die Patientenliege. Damit nach der sequenziellen Abarbeitung dieser Konfiguration keine Szene zurückbleibt, in der nichts mehr passiert, kann die Ausführungsart der Konfiguration angegeben werden. Mögliche Ausprägungen sind die einmalige Ausführung (*Single*) sowie die endlose Wiederholung in einer Schleife (*Loop*). Die Bewegungskonfiguration wird in mehrere Konfigurationen für den Linearbeschleuniger und exakt gleich viele Konfigurationen für die Patientenliege aufgeteilt, da diese, auch wenn sie in der Regel zusammenwirken, autonom voneinander sind.

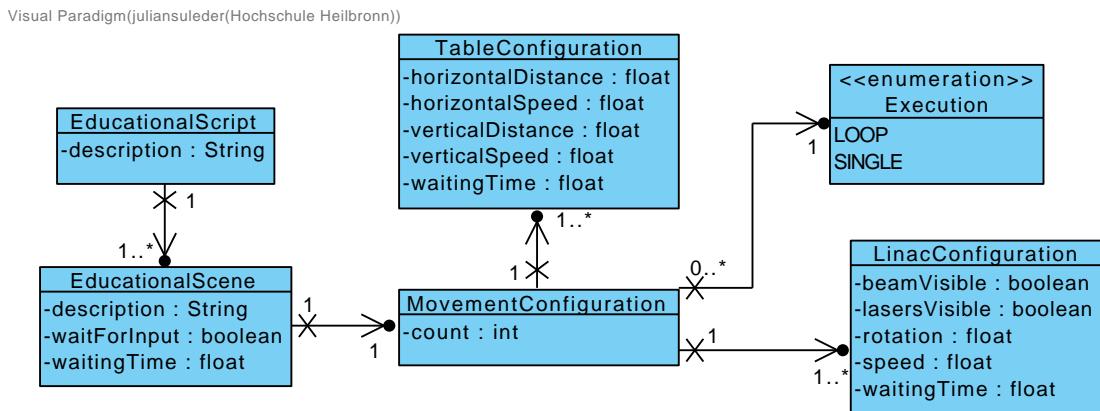


Abbildung 4.12: UML-Klassendiagramm der Domain des Demonstrators. Eine 3D-Szene besitzt ein Skript, welches aus mehreren filmischen Szenen besteht. Diese Szenen bestehen aus Bewegungskonfigurationen für alle Akteure, die ihre Zustände verändern können sowie einer Ausführungsart.

Die Konfiguration des Linearbeschleunigers besteht dabei aus der Angabe, ob der Strahl und die Positionierungslaser angeschaltet sind (*beamVisible*, *lasersVisible*), der Rotation der Gantry (*rotation*) sowie der dafür verwendeten Geschwindigkeit in Sekunden für eine ganze Umdrehung (*speed*) sowie der Zeit, die nach der Bewegung gewartet werden soll (*waitingTime*). Der Sinn der Wartezeit besteht darin, dass ein weicherer Übergang zwischen verschiedenen Konfigurationen ermöglicht wird. Falls zum Beispiel die Richtung der Bewegung verändert wird, bleibt die Gantry des Beschleunigers für diesen Moment in Ruhe, bevor die neue Konfiguration die Bewegung in die andere Richtung initiiert.

Die Konfiguration der Patientenliege besteht aus den Distanzen für die Bewegung der Segmente der Liege in horizontaler sowie vertikaler Richtung in Zentimetern (*horizontalDistance*, *verticalDistance*). Für beide Bewegungen wird eine Geschwindigkeit angegeben (*horizontalSpeed*, *verticalSpeed*). Diese ist als die für die komplette Bewegung benötigte Zeit definiert. Auch die Konfiguration der Patientenliege besitzt eine Wartezeit. Die Wartezeit ermöglicht nicht nur einen weichen Übergang zwischen den Konfigurationen, sondern erfüllt die Bedingung, dass Patientenliege und Beschleuniger in Ruhe bleiben können, da sich selten beide Akteure gleichzeitig bewegen. Es wird angenommen, dass es gleich viele Konfigurationen für Patientenliege und Beschleuniger gibt, da die Parametrisierung der Konfigurationen einfacher ist, wenn diese miteinander korrespondieren.

### 4.5.4 Parametrisierung von Bewegungsabläufen

Korrespondierende Konfigurationen erhalten meist vorbesetzte Parameter. Beispielsweise sind bei einer Drehung der Gantry alle Parameter der Konfiguration der Patientenliege, außer der Wartezeit, auf 0 gesetzt. Es gibt in der 3D-Szene insgesamt vier vordefinierte Funktionen für häufige Parameterkonfigurationen. Die Bewegung der Patientenliege, die Bewegung der Gantry, die Bestrahlung und das Warten aller Akteure.

Diese Parametrisierungen abstrahieren die Erweiterung der Arrays der Bewegungsabläufe, wie sie in Abbildung 4.13 dargestellt ist. Die Parameter der *Add To Arrays*-Funktion werden mit den *Make*-Knoten zu ihren Konfigurationen zusammengefasst. Ein *Make*-Knoten ist das Gegenstück zu einem *Break*-Knoten und vereint die Pins zu einem komplexeren Datentypen. Anschließend werden die Arrays der einzelnen Konfigurationen um die neuen Konfigurationen erweitert.

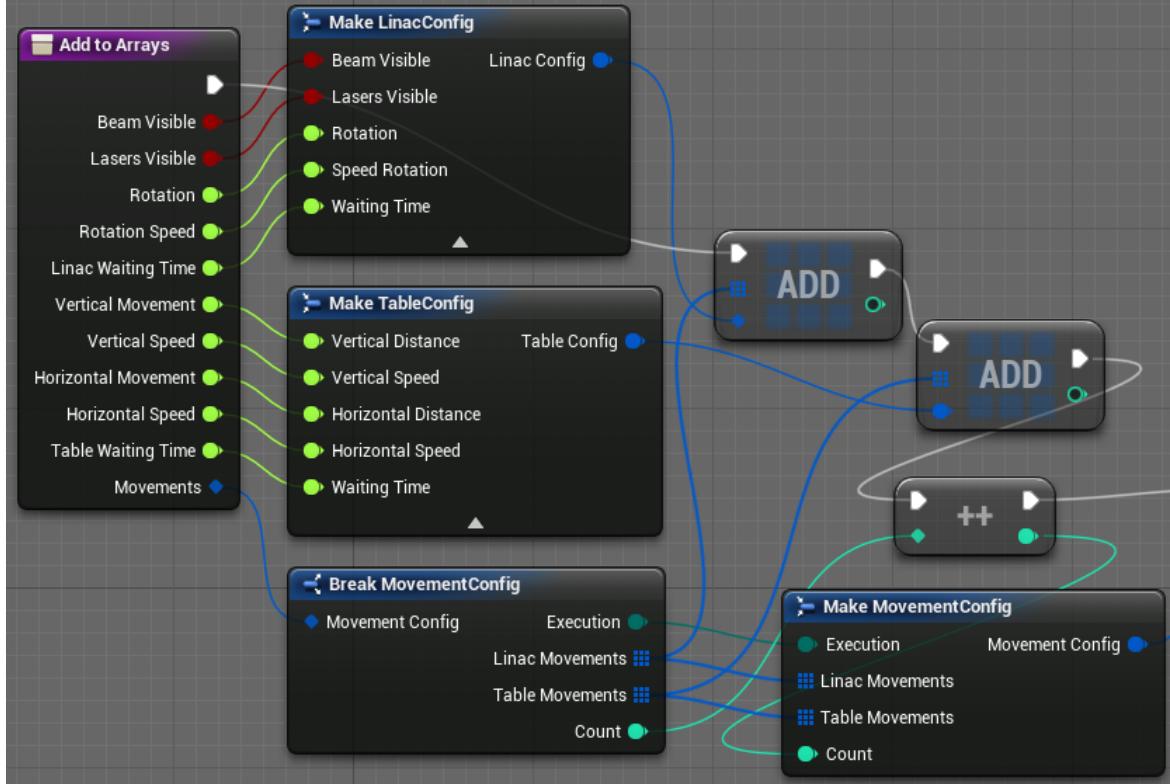


Abbildung 4.13: Der abgebildete Blueprint zeigt das Hinzufügen einer Konfiguration zu den bestehenden Konfigurationen. Durch einen *Make*-Knoten werden Parameter zu einem komplexeren Datentyp zusammengefasst. Die so entstandenen Konfigurationen werden zu den Arrays der bestehenden Konfigurationen hinzugefügt.

In Abbildung 4.14 ist die Funktion für die Parametrisierung der Bewegung der Gantry des Beschleunigers abgebildet, bei welcher die Sichtbarkeit von Laser und Strahl angegeben wird. Die angegebene Rotation wird mit der Geschwindigkeit unverändert in die Konfiguration der Beschleunigers übernommen. Die Gantry bewegt sich bei positiven Winkeln frontal betrachtet im Uhrzeigersinn, bei negativen Winkeln gegen den Uhrzeigersinn. Die Geschwindigkeit muss größer als  $0$  sein. Die Wartezeit der Liege wird durch die Zeit, die die Gantry für die Drehung um den gegebenen Winkel benötigt sowie der anschließenden Pause berechnet. Dafür wird die absolute Drehung durch  $360$  geteilt, um herauszufinden, welchen Bruchteil der Drehung durchgeführt wird. Anschließend wird dieser Bruchteil mit der Geschwindigkeit der Bewegung multipliziert. Das Ergebnis ist die Zeit, welche die Gantry für die Bewegung benötigt. Mit der Pause nach der Bewegung addiert, ergibt dies die Wartezeit der Liege.

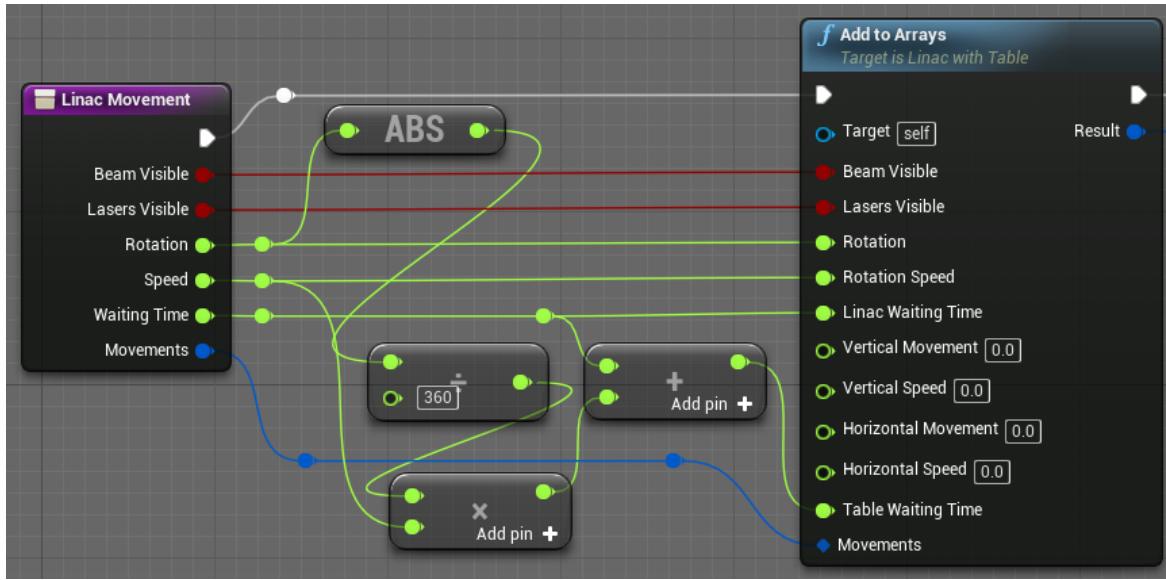


Abbildung 4.14: Der abgebildete Blueprint zeigt die Konfiguration der Bewegung der Gantry. Es kann angegeben werden, ob die Laser und der Strahl sichtbar sein sollen. Zusätzlich wird über die Zeit, die die Gantry benötigt, um die Drehung durchzuführen sowie der anschließenden Pause die Wartezeit der Patientenliege berechnet.

In Abbildung 4.15 ist die Funktion für die Parametrisierung der Bewegung der Segmente der Liege abgebildet. Auch hier kann angegeben werden, ob während der Bewegung die Laser und der Strahl sichtbar sein sollen. Diese gehen, wie die Geschwindigkeiten und Distanzen der Bewegungen der Segmente der Liege, unverändert in deren Konfiguration ein. Dabei bedeuten eine positive horizontale Distanz eine Bewegung in Richtung des Beschleunigers und eine positive vertikale Distanz eine Bewegung nach oben. Äquivalent bedeuten negative Distanzen Bewegungen vom Beschleuniger weg und nach unten. Da es vorkommen kann, dass eine Zeitspanne für die Bewegung angegeben wird, aber keine Bewegung stattfindet, wird zunächst überprüft, ob für jedes Segment eine Bewegung durchgeführt wird und anschließend deren Dauer in einer lokalen Variable gespeichert. Die größere der beiden bildet mit der Pause der Liege die Wartezeit des Beschleunigers.

## 4 Ergebnisse

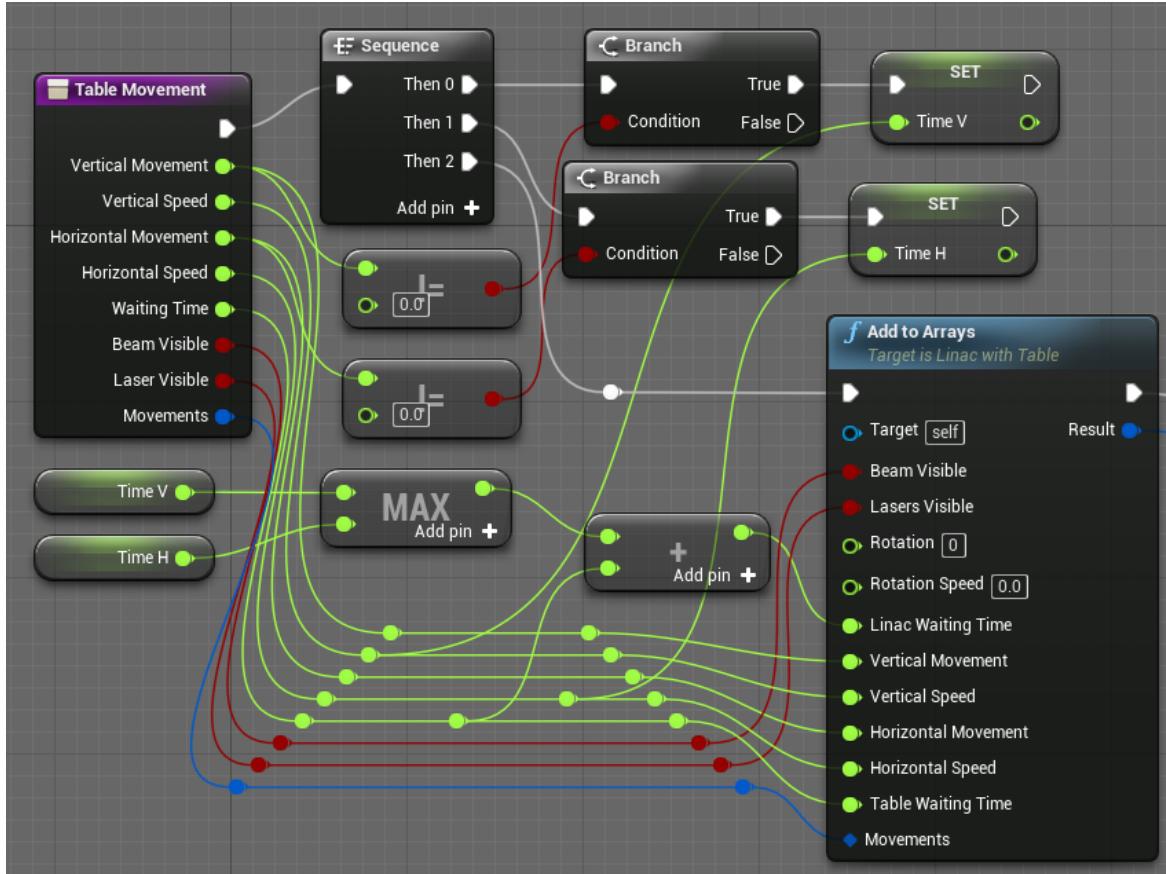


Abbildung 4.15: Der abgebildete Blueprint zeigt die Konfiguration für die Bewegung der Patientenliege. Es wird die Sichtbarkeit von Laser und Strahl angegeben. Zusätzlich wird über die Zeit, die die Segmente für ihre eventuelle Bewegung benötigen sowie die anschließende Pause die Wartezeit der Gantry berechnet.

Für die in Abbildung 4.16 dargestellte Bestrahlung werden zwei Konfigurationen erstellt. Diese unterscheiden sich in der Sichtbarkeit des Strahls und der Wartezeit der Geräte. Die Wartezeit von Liege und Linearbeschleuniger während der Bestrahlung wird als die Dauer der Bestrahlung gesetzt. In der zweiten Bewegungskonfiguration wird diese durch die Pause nach der Bestrahlung definiert. Alle anderen Parameter werden mit  $\theta$  belegt.

## 4 Ergebnisse

---



Abbildung 4.16: Der abgebildete Blueprint zeigt die Konfiguration für die Bestrahlung. Es werden zwei Konfigurationen erstellt, die sich in der Sichtbarkeit des Strahls und der Wartezeit der Geräte unterscheiden. Alle anderen Parameter werden mit  $\theta$  belegt.

In Abbildung 4.17 ist abgebildet, wie die Funktionen aus den Abbildungen 4.14, 4.15 und 4.16 nacheinander parametrisiert werden können, um eine filmische Szene ähnlich wie in einem Drehbuch zu erstellen. Die abgebildeten Konfigurationsfolgen resultieren in der in Tabelle 4.2 angegebenen Array-Belegung. Diese Belegung wird den Aktoren zu Beginn des Levels übergeben und nacheinander abgearbeitet. In der Tabelle sind alle Zellen, die in der Konfiguration angegeben werden, rot markiert. Alle Zellen von berechneten Werten sind blau markiert. Alle anderen Zellen sind durch die Auswahl der Konfiguration, mit Ausnahme des Bestrahlungsstrahls bei der ersten Konfiguration der Bestrahlung,  $\theta$ .

## 4 Ergebnisse

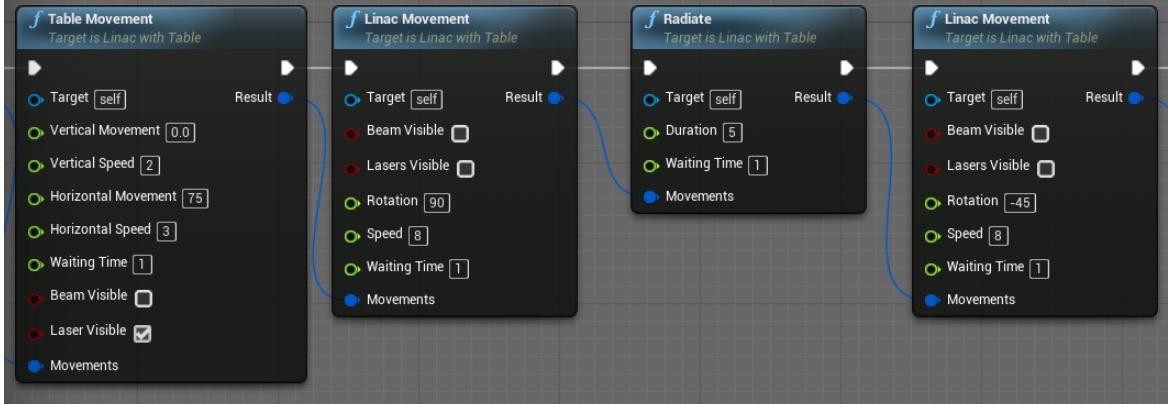


Abbildung 4.17: Der abgebildete Blueprint zeigt eine Konfigurationsfolge aus den in Abbildung 4.14, 4.15 und 4.16 abgebildeten Konfigurationen.

|                           | Table Movement | Linac Movement | Radiate | Linac |
|---------------------------|----------------|----------------|---------|-------|
| Beam                      | 0              | 0              | 1       | 0     |
| Lasers                    | 1              | 0              | 0       | 0     |
| Rotation                  | 0              | 90             | 0       | 0     |
| Rotation Speed            | 0              | 8              | 0       | 8     |
| Linac Waiting Time        | 4              | 1              | 5       | 1     |
| Vertical Movement         | 0              | 0              | 0       | 0     |
| Vertical Movement Speed   | 2              | 0              | 0       | 0     |
| Horizontal Movement       | 75             | 0              | 0       | 0     |
| Horizontal Movement Speed | 3              | 0              | 0       | 0     |
| Table Waiting Time        | 1              | 3              | 5       | 1     |
|                           |                |                |         | 2     |

Tabelle 4.2: Bewegungsfolge, wie sie in den Arrays für die Abarbeitung durch die Akto- ren angelegt wird. Rote Zellen werden in den Konfigurationen gesetzt, blaue berechnet, weiße durch die Wahl der Konfiguration erschlossen.

### 4.5.5 Drehung der Gantry

Für die Drehung der Gantry wird das Event-System der Unreal Engine verwendet. Das bedeutet, dass durch die Blueprints asynchron Funktionen aufgerufen werden können, auch Events genannt. Das eignet sich für das Starten der Bewegung der Gantry, da zu Beginn der Bewegung das Geräusch des Motors gestartet und nach Beendigung der Bewegung wieder gestoppt werden soll. Gleichzeitig muss nicht in jedem Simula- tionsschritt eine Prüfung des Zustands des Linearbeschleunigers stattfinden. Denn die asynchrone Bewegungsinitiierung verändert diesen ebenfalls.

Zu Beginn des Events wird der Zustand des Linearbeschleunigers, sofern die Bewegung ungleich null Grad ist, auf *Bewegend* gesetzt. Anschließend wird mit der bisherigen Transformation der Gantry des Linearbeschleunigers und dem zu rotierenden Winkel eine Zieltransformation berechnet. Daraufhin wird das Geräusch des Motors gestartet und über eine sogenannte Timeline die Bewegung selbst initiiert. Die Timeline ist ein Konstrukt der Unreal Engine, in welcher man asynchron ein Bewegungsprofil definieren kann. Zudem kapselt die Timeline das Berechnen der aktuellen Schrittweite pro Simulationsschritt. Das bedeutet, dass man an dem *Update*-Knoten des Kontrollflusses einer Timeline die in jedem Simulationsschritt durchzuführenden Aktionen spezifiziert. In diesem Fall ist das die Interpolation der Startrotation zur Zielrotation und die Aktualisierung der aktuellen relativen Rotation. Dies funktioniert mit einem sogenannten *Lerp*-Knoten, der eine lineare Interpolation zweier Transformationen berechnet. Als Ausgangstransformation fungiert hierbei die aktuelle Transformation der Gantry, als Zieltransformation die vorherig berechnete Transformation inklusive der neuen Rotation. Die Timeline steuert bei dieser Interpolation die aktuelle Schrittweite, in der Abbildung ist diese als Datenfluss *RotationCharacteristics* dargestellt, und beschleunigt oder verlangsamt diese ihrem Bewegungsprofil entsprechend. Das führt dazu, dass die lineare Interpolation zeitlich verzerrt wird und exakt dem Profil der Timeline entspricht. Nach Abschluss der Bewegung, d.h. wenn die Timeline an ihrem Ende angekommen ist und die Interpolation damit die Zieltransformation erreicht hat, wird der Finished-Kontrollfluss einmalig ausgeführt. Bei der Drehung der Gantry bedeutet dies, dass der Zustand des Linearbeschleunigers zu *nicht bewegend* gesetzt und das Bewegungsgeräusch des Motors beendet wird.

Prinzipiell dauert das Durchlaufen einer Timeline genau eine Sekunde. Damit die Dauer einer Bewegung angepasst werden kann, wird die sogenannte *PlayRate* der Timeline durch die Bewegungsdauer geteilt, d.h. um diesen Faktor verlangsamt. Die in Abbildung 4.18 dargestellte Timeline verzerrt die lineare Interpolation in der Weise, dass die in der Realität schwere Gantry langsam beginnt, sich zu bewegen. Nach einer relativ konstant schnellen Bewegungsphase wird diese entsprechend wieder gebremst. Da der Event asynchron ausgeführt wird, ist es notwendig, die Bewegung stoppen zu können. Dies wird durch einen eigenen Event (*Stop Movement*) realisiert, der das Stoppen der Timeline ausführt. Anschließend wird, wie bei einer regulären Beendigung der Timeline, der Finished-Kontrollfluss ausgeführt.

## 4 Ergebnisse

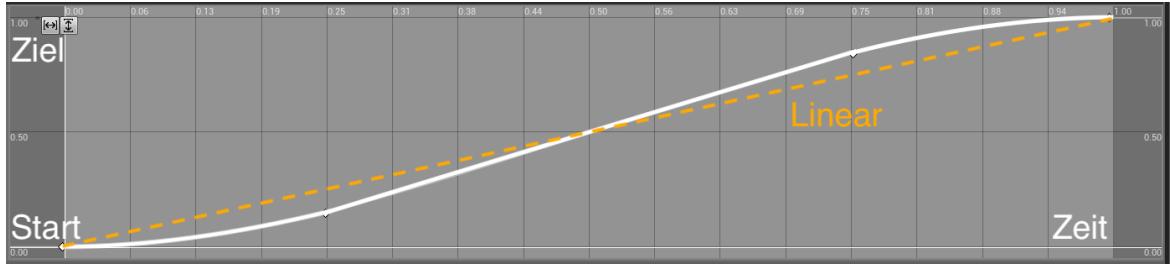


Abbildung 4.18: Die dargestellte Timeline verzerrt die lineare Interpolation in der Weise, dass die in der Realität schwere Gantry langsam beginnt, sich zu bewegen. Nach einer relativ konstant schnellen Bewegungsphase wird diese entsprechend wieder gebremst, was die Trägheit des schweren Objekts plausibel erscheinen lassen soll.

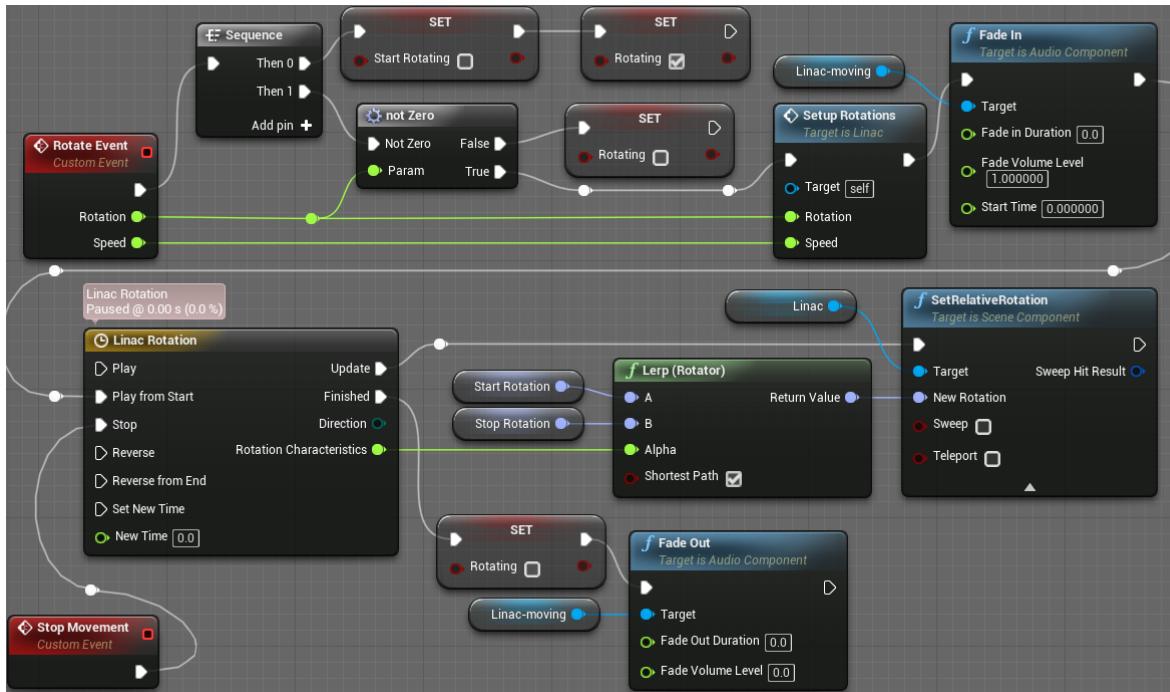


Abbildung 4.19: In der Abbildung ist der Event für das Initiieren der Bewegung der Gantry abgebildet. Nach einer Aktualisierung des Zustands des Linearbeschleunigers wird mit der aktuellen Rotation der Gantry und dem zu drehenden Winkel eine Zielrotation berechnet. Anschließend wird das Bewegungsgeräusch der Gantry gestartet und durch die Timeline die Bewegung für jeden Simulationsschritt gesetzt. Hierfür wird eine Interpolation der Start- und Zielrotationen gemäß des Bewegungsprofils der Timeline durchgeführt. Abschließend wird der Zustand des Linearbeschleunigers aktualisiert und das Bewegungsgeräusch gestoppt.

#### 4.5.6 Bewegen der Patientenliege

Für die Bewegung der Patientenliege wird wie bei der Bewegung der Gantry das Event-System der Unreal Engine verwendet. Da die Bewegungen in horizontaler und vertikaler Richtung durch verschiedene Segmente der Liege durchgeführt werden und daher komplett unabhängig zueinander sind, werden zwei eigenständige Events erstellt. Diese unterscheiden sich nur in der Verwendung der Variablen und des bewegenden Segments, die Funktionsweise ist dieselbe. Diese wird in Abbildung 4.20 dargestellt. Äquivalent zur Bewegung der Gantry wird der Zustand der Patientenliege aktualisiert. Anschließend wird die neue Position des Tischsegments berechnet und eine Timeline für die Interpolation der Ausgangs- und Zielposition des Segments gestartet. Dadurch wird auch hier erreicht, dass der Tisch nicht ruckartige Bewegungen durchführt, sondern eine Beschleunigung und ein Abbremsen stattfinden. Das Aktualisieren der Position des transformierten Segments wird entsprechend des Szenographs auf alle untergeordneten Objekte, d.h. die darauf aufbauenden restlichen Segmente der Liege sowie alle darauf liegenden Objekte oder Patienten angewandt. Um Auch hier die Bewegungen des Tisches stoppen zu können, wird ein Stopping-Event erstellt, der beide Timelines, d.h. horizontal und vertikal, zum Stoppen bringt.

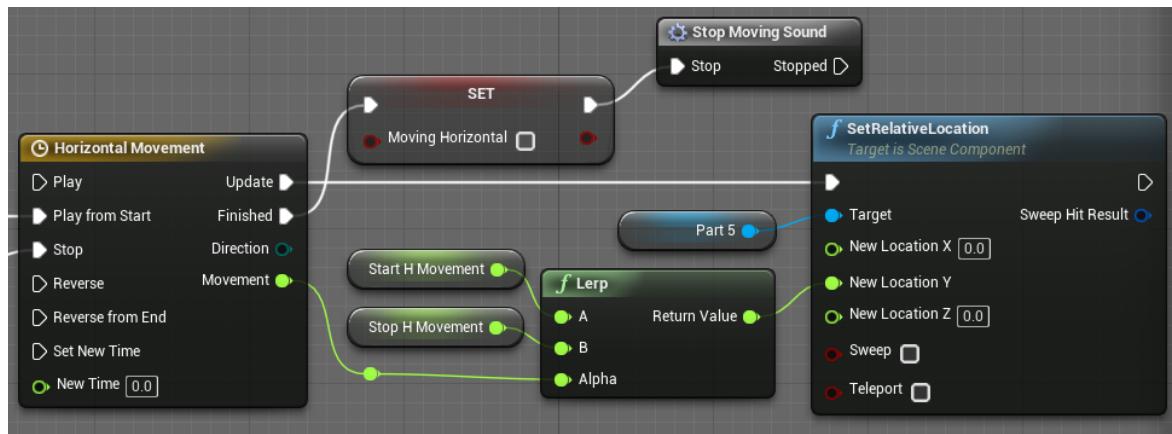


Abbildung 4.20: Die Abbildung zeigt den Event für das mit einer Timeline verzerrte Bewegen des horizontalen Segments der Patientenliege.

#### 4.5.7 Aktives Warten von Aktoren

Das Warten von Patientenliege und Gantry muss synchron durchgeführt werden. Warten bedeutet, dass ein Objekt in diesem Simulationsschritt keine Bewegung oder Animation durchführt. Dazu wird nach jedem neuen Simulationsschritt das Intervall zum alten Simulationsschritt (*Delta Seconds*) zu einem Zähler addiert (siehe Abbildung 4.21). Ist der Zähler größer als die Zeit, die gewartet werden soll, so wird der Zustand des Objekts aktualisiert und der Zähler zurückgesetzt.

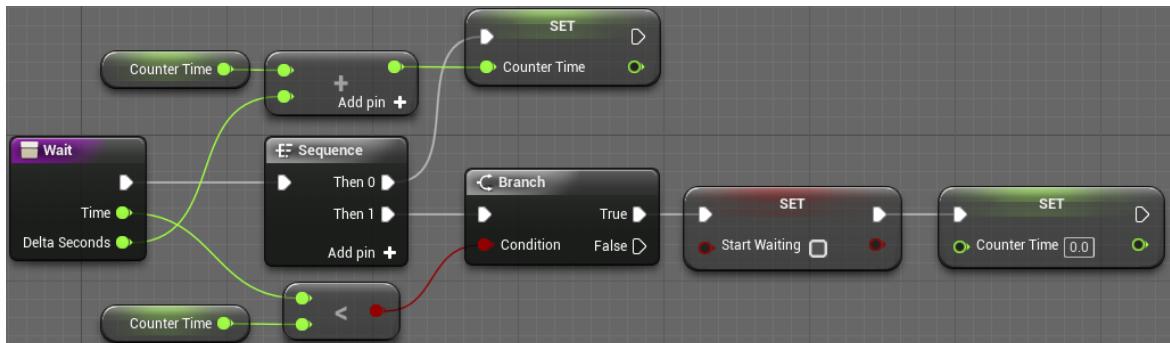


Abbildung 4.21: Aktualisierung des Zählers für das Warten von Simulationsmodellen. Am Ende der Wartezeit erfolgt ein Zustandswechsel des Beschleunigers.

#### 4.5.8 Zustände des Linearbeschleunigers

Es gibt für den Linearbeschleuniger vier Zustände zwischen denen er in der Simulation wechselt. Diese sind *Bewegung Starten*, *Bewegen*, *Warten* und *Fertig*. In Abbildung 4.22 ist der Tick-Event des Linearbeschleunigers dargestellt, welcher, wie in Abschnitt 3.5.2.2 vorgestellt, in jedem Simulationsschritt ausgelöst wird. In der Funktion *Split Movement* wird festgestellt, in welchem Zustand sich der Linearbeschleuniger aktuell befindet (siehe Abbildung 4.23). Im Zustand *Bewegung Starten* wird der Event für die Gantry-Bewegung ausgelöst, sodass der Linearbeschleuniger in den Zustand *Bewegen* wechselt. In diesem Zustand wird asynchron die Bewegung durch den in Abschnitt 4.5.5 vorgestellten Event gestartet. In den nachfolgenden Simulationsschritten befindet sich der Linearbeschleuniger immer noch im Zustand *Bewegen*, da der Event der Bewegung noch nicht vollständig abgearbeitet ist. Erst am Ende des Events wechselt der Linearbeschleuniger in den Zustand *Warten*. In diesem Zustand wartet der Linearbeschleuniger, wie in Abschnitt 4.5.7 beschrieben. In jedem der Zustände *Bewegung Starten*, *Bewegen* und *Warten* wird der *Finally*-Kontrollfluss ausgeführt. Mit diesem Kontrollfluss wird die Sichtbarkeit der Laser und des Bestrahlungsstrahls sowie der Statusanzeige aktualisiert. Zusätzlich werden, sofern das Warten beendet wurde, der Index der Bewegungsfolge inkrementiert und der Linearbeschleuniger in den Zustand *Bewegung Starten* gebracht. Durch diese Überprüfung in jedem Simulationsschritt ist

es möglich, dass der Linearbeschleuniger nacheinander die in Abschnitt 4.5.4 angegebenen Bewegungsabläufe abarbeiten kann. Dies kann nicht immer beim Beenden des Wartens durchgeführt werden, sondern muss nach jedem Simulationsschritt geprüft werden, da es Bewegungskonfigurationen geben kann, in welchen gar nicht gewartet werden soll und der Linearbeschleuniger dann in einen inkonsistenten Überstand übergehen würde. Die Art der Ausführung fließt in die Überprüfung mit ein. Wenn die Bewegungsfolge abgearbeitet ist, ist der Index gleich der Anzahl der Bewegungskonfigurationen. Mit der Art der Ausführung wird entschieden, ob der Index erneut auf 0 zurückgesetzt werden soll. Dies führt dazu, dass ein neues Durchlaufen der Bewegungsfolge erfolgt. Andernfalls wird der Linearbeschleuniger in den Zustand *Fertig* gebracht.

Der Blueprint der Patientenliege besitzt vergleichbare Zustände. Im Zustand *Bewegung Starten* werden statt einer Rotation die Bewegungen der beiden Segmente ausgelöst. Dabei erfolgt der Übergang in den Zustand *Warten* erst, wenn beide Rotationen vollständig abgeschlossen sind.

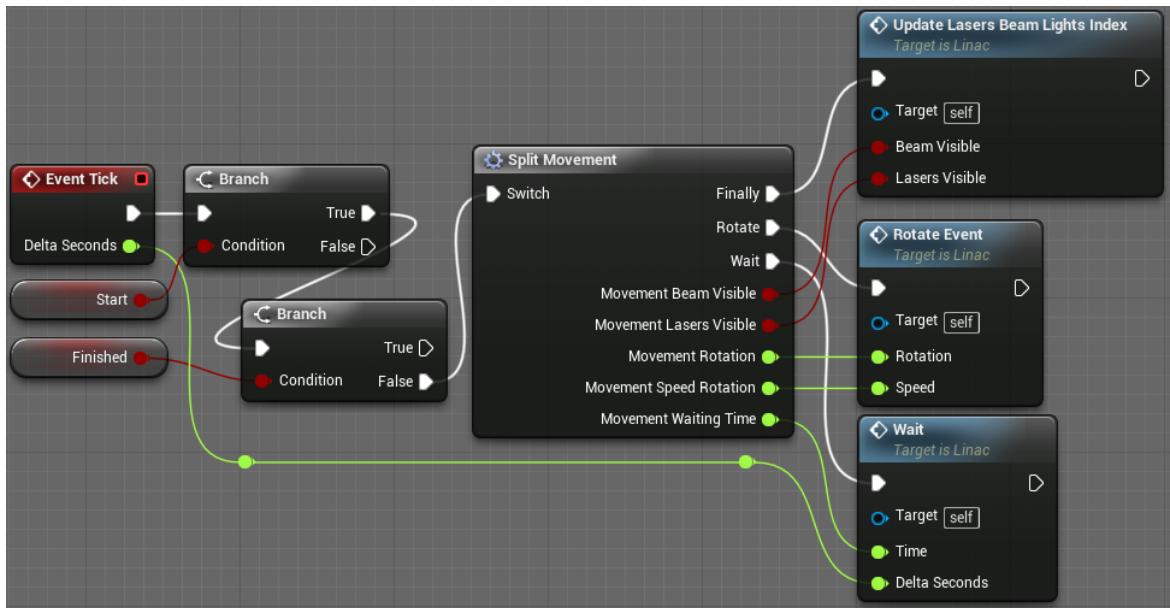


Abbildung 4.22: Der Tick-Event des Linearbeschleunigers prüft den Zustand des Linearbeschleunigers durch die in Abbildung 4.23 abgebildete Funktion in jedem Simulationsschritt. An die mit dem Zustand verknüpfte Aktion folgt immer ein *Finally*-Block, der zusichert, dass der Linearbeschleuniger in einem konsistenten Zustand bleibt.

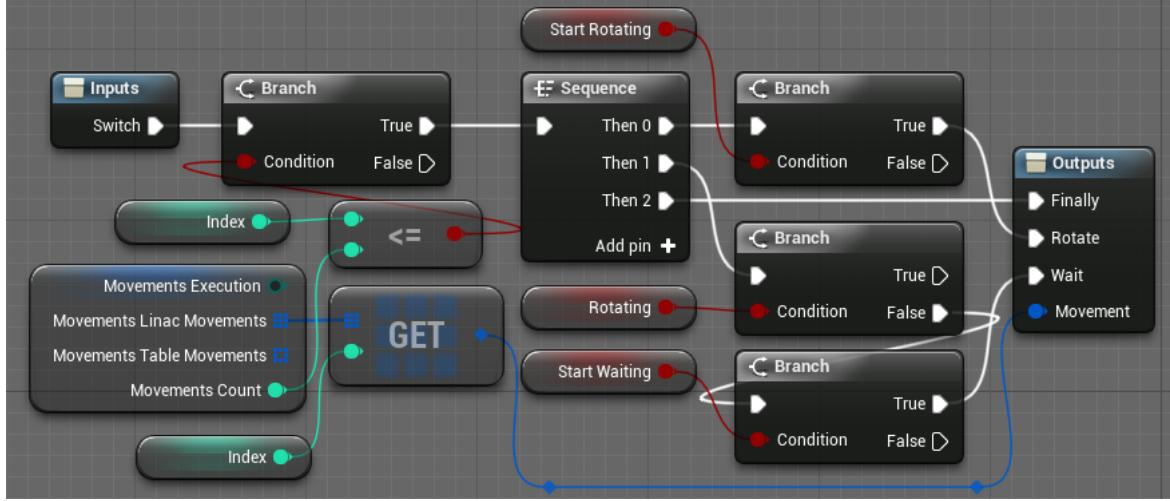


Abbildung 4.23: Die abgebildete Funktion prüft, in welchem Zustand sich der Linearbeschleuniger befindet, und führt den Kontrollfluss des Zustands aus. Anschließend wird immer der *Finally*-Kontrollfluss ausgeführt.

#### 4.5.9 Realisierung von Bestrahlungsstrahl und Laser durch ein Partikelsystem

Für die Realisierung eines Lasers wird das Partikelsystem der Unreal Engine verwendet. Es ermöglicht die Simulation von Lichtstrahlen durch Partikel, die in ihren Eigenschaften, wie z.B. dem Erscheinungsbild durch ein Material, der Lebensdauer der Partikel, Distanz des Strahls und Streuung, angepasst werden können. Im Rahmen des Demonstrators wird der Strahl durch einen grünen Laserstrahl realisiert, die Positionierungslaser sind rot. Für beide Laser wird dasselbe Partikelsystem als Vorlage verwendet, für welches ein Start- und ein Endpunkt angegeben werden müssen. Ein roter Positionierungslaser ist in Abbildung 4.25 dargestellt. Das Partikelsystem erzeugt einen kontinuierlichen Fluss zwischen diesen Punkten im Raum in Form eines Materials. Dieser Fluss kann als eine plane Fläche gesehen werden, deren Normale stets in Richtung der Kamera, d.h. Position des Betrachters, ausgerichtet ist. Das Material ist eine 2D-Textur, die in horizontaler und vertikaler Richtung auf den Strahl gestreckt wird. In Abbildung 4.24 ist die Berechnung des Materials dargestellt. Die finale Textur wird aus zwei einzelnen Texturen berechnet. Dafür werden zwei Texturen verwendet, die durch Addition und zyklische Verschiebung in jedem Simulationsschritt mit einem sogenannten Panner ein Pulsieren erzeugen. Das Ergebnis der Addition ist in jedem Simulationsschritt eine Überlagerung beider Texturen. Die erste der beiden Texturen ist in der Abbildung mit *Textur A* beschriftet und sorgt für die Fokussierung des Strahls in vertikaler Richtung sowie für ein zusätzliches Streuen außerhalb des Zentrums des Strahls. Die zweite Textur ist in der Abbildung mit *Textur B* beschriftet und sorgt mit dem Panner für das Pulsieren des Strahls, da sie bei der Addition

der beiden Texturen in einer unterschiedlichen Gewichtung in vertikaler Richtung abhängig von der horizontalen Verschiebung durch den Panner resultiert. Das Ergebnis der Addition wird mit der Farbe und Transparenz des individuellen Partikels multipliziert. Daraus resultiert die Farbe bzw. Transparenz der entstehenden Textur abhängig für jeden Partikel. Dies führt dazu, dass ein Partikel, der weiter weg von der Mitte des Strahls lokalisiert ist, transparenter dargestellt wird, als ein Partikel in der Mitte des Strahls. Eine zunehmende Breite des Partikelsystems führt zu einer Verkleinerung des Fokus des Strahls. Das Partikelsystem besitzt einen Start- und einen Endpunkt, zwischen welchen der Strahl dargestellt wird. Die Positionierung dieser Punkte wird im nachfolgenden Kapitel 4.5.10 dargelegt.

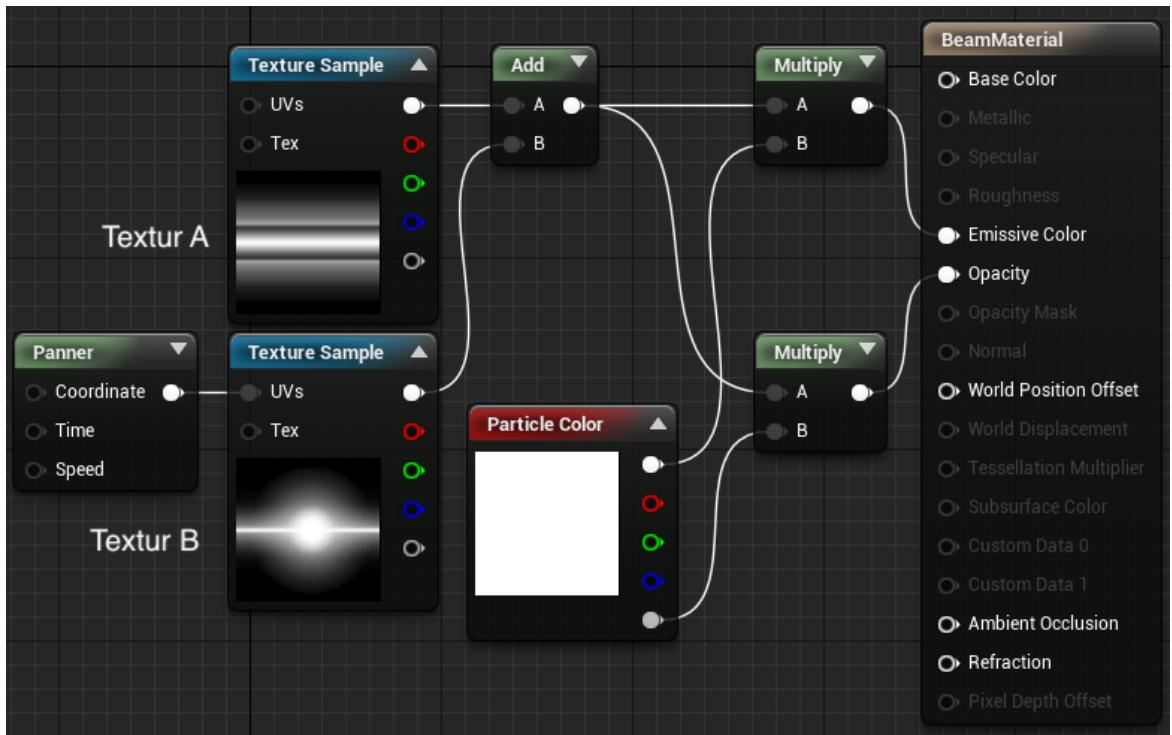


Abbildung 4.24: Das Material des Bestrahlungsstrahls besteht aus zwei Texturen, die in jedem Simulationsschritt übereinander verschoben werden, was zu einem Pulsieren führt. Multipliziert mit der Partikelfarbe und dessen Transparenz, entsteht ein Strahl, der fokussiert ist und um sein Zentrum charakteristisch leuchtet.

Das Partikelsystem ist atomar, d.h. es kann nicht in Strahlenbündel oder Ähnliches aufgeteilt werden. Bei Kollisionen wird diese Eigenschaft in der Simulation relevant, da keine Kollimation des Partikelsystems möglich ist. Das Einschränken durch Blenden führt dazu, dass das komplette Partikelsystem an der Blende endet. Ein Ansatz mit den Mitteln der Engine ist es, das Partikelsystem durch ein Mesh zu ersetzen. Nimmt man statt dem Partikelsystem einen Quader und lässt dessen Tiefe äquivalent zur Spur-

verfolgung variieren, so ergibt sich dass Problem, dass nur ein Berührpunkt auf beliebig geformten Objektoberflächen berechnet wird. Das führt dazu, dass der Rand des Quaders vor der Oberfläche des Objekts endet oder in dieses hineinragt (siehe Abbildung 4.26). Eine Kollimation des Quaders ist trotzdem nicht möglich. Dazu muss der Quader in viele kleinere Quader zerlegt werden. Das Problem, dass deren Rand vor der Oberfläche des Objekts endet oder in dieses hineinragt, reduziert sich auf das Überlappen der Teilquader (siehe Abbildung 4.26). Allgemein bewirkt die Zerlegung, dass die Quader die Oberfläche ähnlich zu einem Oberflächenintegral approximieren. Nachteil des Ansatzes ist, dass die Kollisionsberechnung eine rechenintensive Operation ist und bei Teilung des Strahls in einen Quader von 8x8 Teilquadern bereits 64 Kollisionen pro Simulationsschritt berechnet werden müssen. Ohne weitere Optimierungen verlangsamt dieser Ansatz die Simulation zu sehr, weshalb er im Rahmen dieser Arbeit nicht verfolgt bzw. realisiert wurde.

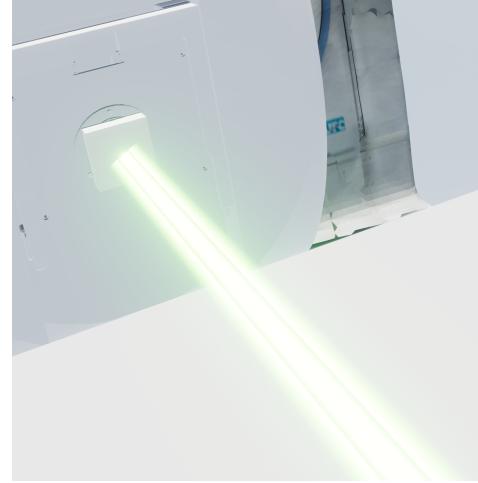


Abbildung 4.25: Der Bestrahlungsstrahl in der VR-Anwendung ist durch ein Partikelsystem realisiert. Die Textur des Bestrahlungsstrahls pulsiert und leuchtet außerhalb des Strahlzentrums.

Arbeit nicht verfolgt bzw. realisiert

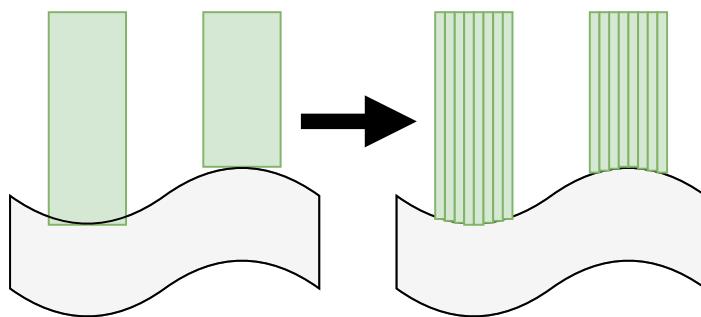


Abbildung 4.26: Die Berechnung der Kollision des Strahls führt zu einem Berührpunkt auf beliebig geformten Objektoberflächen. Der Rand des Quaders endet vor der Oberfläche des Objekts oder ragt in dieses hinein. Eine Zerlegung des Quaders reduziert diese Effekte und führt zu einer rechenintensiven Approximation der Oberfläche.

#### 4.5.10 Spurverfolgung bei Laser und Bestrahlungsstrahl

Die beiden Positionierungslaser und der Bestrahlungsstrahl werden durch das in Abschnitt 4.5.9 beschriebene Partikelsystem realisiert. Der Startpunkt der Partikelsysteme ist immer an ein Mesh eines Aktors gebunden. Durch die relative Positionierung an dessen Szenengraph wird der Startpunkt automatisch aktualisiert. Der Endpunkt des Strahls bzw. Lasers muss in jedem Simulationsschritt neu berechnet werden. Die Berechnung wird mit einer Spurverfolgung, ein sogenanntes *Line Trace*, durchgeführt und ist in Abbildung 4.27 dargestellt. Für das Line Trace wird von einem Startpunkt bis zu einem Endpunkt eine Kollision mit einem Mesh gesucht. Hierfür wird ein auf dem Mesh des Aktors platzieter Vektor (*Laser Start Point*) verwendet. Dieser Vektor steht senkrecht auf der Oberfläche des Aktors und definiert die Ausbreitungsrichtung des Partikelsystems. Der Vektor wird durch Multiplikation so weit verlängert, dass er die komplette virtuelle Welt abdecken kann. Der Endpunkt der Spurverfolgung wird anschließend durch Addition der Position des Vektors mit seiner verlängerten Richtung berechnet und vorerst als Endpunkt des Partikelsystems gesetzt. Das Partikelsystem erstreckt sich vom Startpunkt vorerst ungeachtet von Kollisionen mit Objekten bis zur berechneten Position. Die Spurverfolgung wird in Unreal Engine durch einen *Line Trace By Channel*-Knoten realisiert. Dieser liefert bei einer Kollision den Berührungs- punkt des getroffenen Meshs zurück, welcher den neuen Endpunkt des Partikelsystems repräsentiert. Dies wird als Kürzung der Ausbreitung des Partikelsystems auf den Teil bis zur ersten Kollision interpretiert.

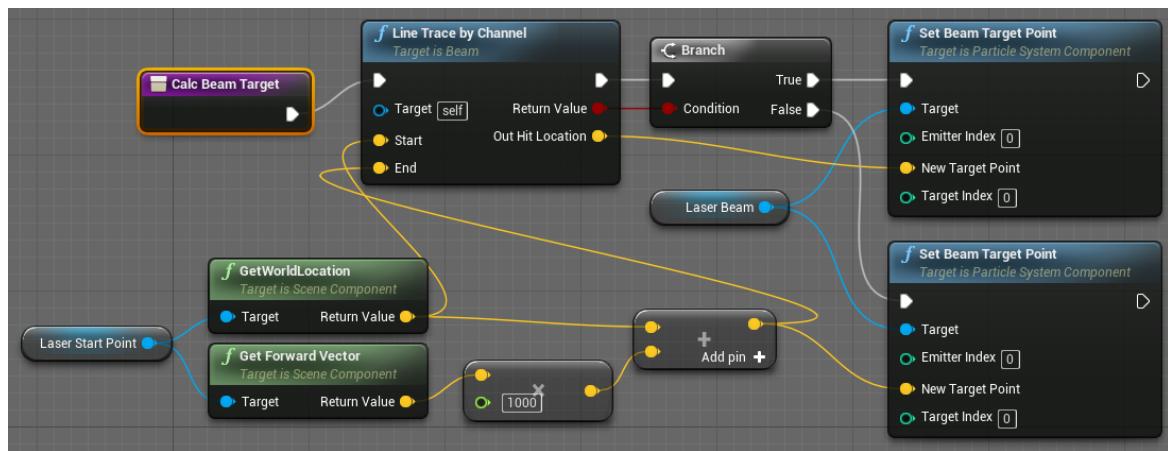


Abbildung 4.27: Die Abbildung zeigt den Blueprint der Spurverfolgung bei Laser und Bestrahlungsstrahl. Der Richtungsvektor des *Laser Start Point* wird verlängert und bildet von seiner Position in der Welt aus den vorläufigen Endpunkt des Partikelsystems. Durch einen *Line Trace By Channel*-Knoten wird dieser durch den Berührungs- punkt der ersten Kollision ersetzt.

#### 4.5.11 Objekt- und Hintergrundgeräusche

Mit der Unreal Engine können nur Audio-Dateien im WAV-Format importiert werden. Andere Dateiformate, wie z.B. MP3, müssen in dieses Format umgewandelt werden. Parameter der Audio-Dateien, wie z.B. Lautstärke, werden mit einem Mixer direkt in der Unreal Engine manipuliert. Für Objektgeräusche werden die Ambient Sound Aktoren, welche die Audiospuren repräsentieren, direkt in den Szenengraphen eines Aktors eingebunden. Dadurch ist die Quelle des Sound Aktors relativ zum Aktor positioniert und ermöglicht, wie in Abschnitt 3.5.2.1 beschrieben, ein Positionieren von Geräuschen in einem Level. Diese Geräusche erscheinen dann in Abhängigkeit von der Distanz zum Aktor lauter oder leiser. Dies wird in den Eigenschaften des Sound Aktors eingestellt. Diese Art von Objektgeräusch wird für die Bewegung der Patientenliege, die Bewegung der Gantry sowie den Bestrahlungsstrahl verwendet. Die Kühlung des Linearbeschleunigers konnte im Bestrahlungsraum als Hintergrundgeräusch keinem direkten Objekt zugeordnet werden. Daher wird hierfür kein Ambient Sound Aktor in Szenengraphen des Linearbeschleunigers aufgenommen. Stattdessen wird das Geräusch in den Szenengraphen des Levels eingefügt. Zusätzlich wird in dessen Einstellungen die Eigenschaft der Lokalisierung des Geräusches deaktiviert. Das hat zur Folge, dass das Geräusch an jeder Position im Raum in derselben Lautstärke wahrnehmbar ist. Die Ambient Sound Aktoren werden durch *Fade In*-Knoten gestartet und durch *Fade Out*-Knoten, wie in Abbildung 4.28 dargestellt, als Aufruf eines Kontrollflusses gestoppt.

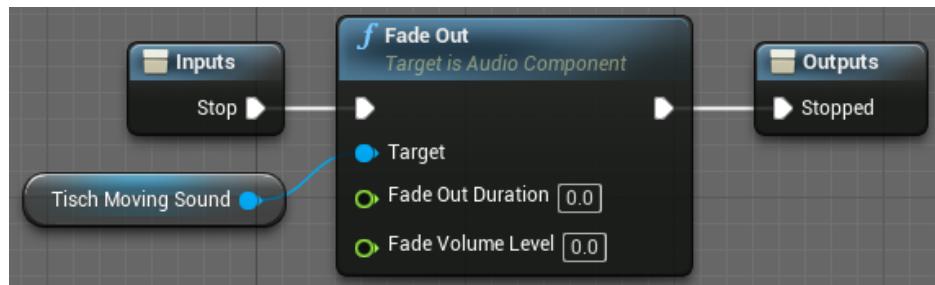


Abbildung 4.28: Der Ambient Sound Aktor wird durch eine Referenz und einen *Fade Out*-Knoten als Kontrollfluss gestoppt.

## 5 Diskussion

Die Ziele dieser Arbeit bestanden aus der Konzeption eines Verfahrens für die Digitalisierung von realen Objekten in der Strahlentherapie unter Berücksichtigung des Anwendungsgebiets und bereits vorhandener Virtual und Augmented Reality Hardware. Dafür wurden Faktoren, welche die Aufklärung und Information von Patienten in der Strahlentherapie beeinflussen können analysiert. Weiterhin sollte das Verfahren in einem Bestrahlungsraum des Deutschen Krebsforschungszentrums in Heidelberg evaluiert und ein Demonstrator entwickelt werden.

In diesem Kapitel werden eine Interpretation und Bewertung der Ergebnisse aus Kapitel 4 durchgeführt. Es ist zu beachten, dass die einzelnen Schritte des Verfahrens zum 3D-Scannen von Objekten aus Abschnitt 4.2 bereits teilweise interpretiert wurden. Dies war notwendig, da die Durchführung weitere Schritte des Verfahrens auf Ergebnissen eines vorhergehenden Schritts basierte.

### 5.1 Konzept zur Patientenaufklärung mit Virtual Reality

In Abschnitt 2.2 wurden Veröffentlichungen im Bereich Patienteninformation und Aufklärung sowie die dafür verwendeten Medien vorgestellt. Es wird bereits eine umfassende Information der Patienten durchgeführt (siehe 2.2), in welcher meist auf standardisierte Materialien, wie Broschüren oder seltener auch Videos und Simulationen, zurückgegriffen wird. Das Konzept sieht vor, in der Strahlentherapie verwendeten Geräte zu digitalisieren und animieren. Die Rolle des Patienten in der Simulation soll nahe an seine reale Rolle angelegt sein. Es sollte eine Einweisung und Erläuterung der Interaktion und des zu erklärenden Konzepts durch einen O-Ton oder das medizinische Personal vorhanden sein.

Es gibt wenige Erkenntnisse in diesem Bereich bezüglich des Einsatzes der neuen Technologien, wie z.B. Virtual Reality und Augmented Reality. Dies liegt unter anderem daran, dass diese den Schritt in den Consumer Markt noch nicht geschafft haben und hauptsächlich in Forschung und Entwicklung getestet werden. Das Prinzip hinter dem Konzept ist, dass der Patient seine Angst in der Simulation abbaut und die reale Szene als vertraut wiedererkennt. Dies wird mit Virtual Reality bereits in der Konfrontationstherapie angewandt, um z.B. Höhenangst zu therapieren (siehe Abschnitt 2.5). In der Medizin kann dies eine Untersuchung mit einem Magnetresonanztomographen sein, bei welchem dem Patienten durch das Head-Mounted-Display und Kopfhörer der Eindruck erweckt wird, in der engen Röhre zu liegen und die typischen Klopft- und Brummgeräusche zu hören (vgl. Abschnitt 2.1.4.2). Durch die Demonstration dieser Szene kann es sein, dass der Patient weniger Angst entwickelt. Auch eine Visualisierung einer Szene, die der Patient in der Position eines Beobachters erlebt, ist denkbar,

um die Mitwirkung des Patienten zu fördern. Hier könnte man in einer Szene die Positionierung des Patienten am Linearbeschleuniger mit anschließender Bestrahlung mit Gating zeigen. Der Patient könnte in einer solchen Szene lernen, durch die Kommandos die Luft lange genug anzuhalten. Erfährt der Patienten die Szene aus seiner eigenen Rolle, ist dabei wichtig, dass diese möglichst nahe an seine reale Rolle angelehnt ist und sich die Präsentation der Inhalte auf die wesentlichen Aspekte der zu erklärenden Szene beschränkt. Ansonsten könnte der Patient zu sehr abgelenkt sein. Dafür sollten bewusst irrelevante Aspekte und Objekte einer Szene weggelassen werden. Zudem sollte eine Szene nur jeweils ein Konzept erklären oder eine Art der Mitwirkung des Patienten trainieren, sodass diese miteinander individuell für jeden Patienten kombiniert werden können. Die Szene sollte so entworfen sein, dass keine große Bewegung notwendig ist, um die Objekte und ihre Funktionen erforschen zu können. Gleichzeitig sollte der Patient unabhängig von seiner realen Position im Raum immer einen Blickwinkel in der Szene einnehmen, durch welchen er alle Geschehnisse überblicken kann. Hierfür sollte, falls nötig, vom Konzept des *Willentlichen Ausblendens des Unglaubens Gebrauch* gemacht werden (Abschnitt 2.3.4). Zudem sollte die Interaktion des Patienten gering gehalten werden, da in der Krebstherapie häufiger ältere als junge Menschen betroffen sind und viele den Umgang mit der neuen Technologie nicht gewohnt sind. Sollte eine Interaktion trotzdem notwendig sein, sollte diese durch das Personal oder einen O-Ton erklärt werden. Die Effekte einer Aufklärung mit nach diesem Konzept entwickelten Simulationen müssen untersucht werden, um genauere Aussagen treffen zu können.

## 5.2 Verfahren zur Digitalisierung von Strahlentherapie-Szenen

In dem in Abschnitt 4.2 vorgestellten Verfahren zur Digitalisierung von Objekten wurde ein Tablet mit spezieller Sensorik verwendet, um mit frei verfügbarer Software 3D-Modelle zu erstellen. Da das Verfahren selbst und seine Methodik bereits bei der Darlegung in den Unterabschnitten interpretiert wurde, erfolgt in den nachfolgenden Abschnitten eine Bewertung dieses Verfahrens und seiner Bestandteile.

### 5.2.1 Das Vergleichsobjekt

Der Computertomograph wurde als Vergleichsobjekt für die Rekonstruktionen ausgewählt, das seine Form zum einen Kanten, zum anderen auch Rundungen besitzt (siehe Abschnitt 4.2.4.1). Seine Oberfläche ist in der Realität matt, was dazu führt, dass durch die Beleuchtung im Bunker des Linearbeschleunigers Reflexionen auf seiner Oberfläche durch die verschiedenen Lichtquellen auftreten, die das Mapping der Textur erschwert haben. Seine Oberfläche besitzt bis auf kleinere Beschriftungen und Logos überall dieselbe Farbe. Daran konnte die Qualität der Textur nach der Rekonstruktion gerade im Bezug auf die automatische Korrektur der Helligkeit der einzelnen

Fotos beurteilt werden. Weitere Gründe für die Auswahl des Computertomographen waren seine Größe und Zugänglichkeit. Der Computertomograph ist auf Schienen gelagert und hätte für das Scannen von der hinter ihm gelegenen Wand weg bewegt werden können. Dies wurde nicht gemacht, da so die Problematik im Anwendungsgebiet des Verfahrens simuliert wurde. Häufig sind die Geräte schwer oder baulich so groß, dass sie nicht aus allen Seiten erfasst werden können. Das erzwingen dieses Sachverhalts beim Vergleichsobjekt ermöglichte den Test des Umgangs des Verfahrens mit diesen Punkt wolken.

### 5.2.2 Qualität der Scans mit dem Tablet

In Abschnitt 4.2.2 wurde das Verfahren zum Scannen mit dem Tablet vorgestellt. Die Genauigkeit der Scans hängt von der Beschaffenheit der gescannten Objekte ab. Feine Strukturen und Kanten können durch die Scans mit dem Tablet nicht erfasst werden, da die Genauigkeit des Scans durch die Hardwarebeschränkungen des Tiefensensors limitiert ist. Bis zu einer Auflösung von fünf Zentimetern können getrennte Objekte durch die Dichte der Punkt wolke als getrennte Objekte abgebildet werden. Darunter verwischt die gemessene Tiefe zu sehr. Für einzelne Kanten eines Objekts gelten diese Einschränkungen in der Art nicht, da kein Verwischen mit anderen Objekten möglich ist. Dort kann die Kante durch mehrfaches partielles Scannen und das durchgeführte Sparse-Bundle Adjustment abgebildet und rekonstruiert werden. Der Tiefensor erkennt dabei Strukturen mit einer Abweichung von etwa einem ganzen Zentimeter. Probleme bilden transparente Flächen und Objekte, die durch das Messprinzip des optischen Tiefensors nicht adäquat erfasst werden können, sowie stark reflektierende Objekte, wie Spiegel oder direkt angeleuchtete, helle Oberflächen. Diese Objekte sollten manuell modelliert werden. Ein Vorteil des Scannens mit dieser App ist, dass, wenn die App zwischen den Scans nicht geschlossen wird, auch ohne aktives Scannen der Pfad des Tablets im Raum erfasst wird. Das führt dazu, dass beim Scannen verschiedener einzelner Objekte in einem Raum die relative Position dieser Objekte zueinander in der Rekonstruktion gleich bleibt, da alle Datensätze nach der anfänglichen Initialisierung in diesem Referenzkoordinatensystem gespeichert werden. Dadurch wird es möglich, einzelne Seiten von Objekten genauer in einzelnen Datensätzen zu scannen und diese im Nachgang zusammenzusetzen. Der Arbeitsspeicher des Tablets ist bei hoher Auflösung der limitierende Faktor für die Größe eines Scans. Durch das Referenzkoordinatensystem sollte ein Scan von einem größeren Objekt daher auf mehrere überlappende Scans aufgeteilt werden. Diese müssen anschließend voneinander getrennt rekonstruiert und zusammengesetzt werden.

### 5.2.3 Auswahl und Parametrisierung der Rekonstruktion

Die Auswahl der Rekonstruktionsalgorithmen beschränkt sich auf die in RTAB-Map implementierte Screened Poisson Surface Reconstruction und die Greedy-Projection Triangulation. Es existieren eine Vielzahl weiterer Algorithmen, die unter Anderem in Meshlab implementiert sind. Bei unzureichenden Ergebnissen der ausgewählten Screened Poisson Surface Reconstruction sollte gegebenenfalls eine Evaluation dieser Algorithmen durchgeführt werden, um bessere Ergebnisse erzielen zu können. Die Parametrisierung der Screened Poisson Surface Reconstruction wurde in Abschnitt 4.2.4 vorgestellt. Es bleibt zu beachten, dass diverse andere Parameter, wie z.B. Dichte der Punktwolke, ob diese geglättet wurde und die Mindestanzahl der Punkte pro Knoten, individuell auf jedes Modell angepasst werden sollten. Die Mindestanzahl der Punkte pro Knoten sollte bei Scans von kleineren Objekten erhöht werden, da sie eine robustere Extraktion von Kanten durch implizite Filterung der Punktwolke ermöglicht. Allgemein kann die Feinheit der Rekonstruktion, d.h. die Größe der Polygone sowie die Glättung durch Hierarchisierung durch die minimale Tiefe des Octrees bei der Rekonstruktion angepasst werden. Die Tiefe der Rekonstruktion kann als dichteabhängige Homogenisierung der Punktwolke angesehen werden. Dichtere Bereiche werden mit der Mindestanzahl von Punkten pro Knoten feiner aufgegliedert als weniger dichte Bereiche, sodass dort feinere Kanten rekonstruiert werden. Gleichzeitig sollte die Homogenisierung nicht durch eine zu hohe minimale Octree-Tiefe verhindert werden, da diese zu höherem Rauschen durch geringeres Glätten der Punktwolke führt. Wie Abbildung 4.4c im Vergleich zu Abbildung 4.4b zeigt, verschlechtert dies trotz deutlich höherer Anzahl an Vertices und Faces den visuellen Eindruck der Rekonstruktion.

### 5.2.4 Bewertung und Probleme des Textur-Mapping

Im Versuch mit dem Head-Mounted Display und der Auflösung der verbauten Displays war die Textur des in Abbildung 4.6 abgebildeten Models einer Wand mit Postern der Größe  $5 \times 3 \text{m}^2$  bereits so scharf, dass man auch bei direkter Betrachtung große Schrift lesen kann. In der Abbildung wird an den mit einem orangefarbenen Pfeil markierten Bereichen ersichtlich, dass in den überlappenden Kamerabildern, Verschiebungen entstehen können, falls die Bilder nicht korrekt ausgerichtet werden. Zudem sieht man in der Abbildung an den mit einem lila farbenen Pfeil markierten Stellen, dass es Bereiche geben kann, die durch die unterschiedliche Ausleuchtung bei der Aufnahme dunkler als ihre Umgebung erscheinen. Mit grünen Pfeilen sind Stellen markiert, die durch die Textur gut lesbar sind. Das sind zum einen Überschriften, bei welchen die Kanten der Schrift kontrastreich verlaufen. Zum anderen sind das Diagramme, in denen trotz der Auflösung der Textur Linien erkennbar sind. Ein Beispiel dafür ist das Diagramm mit der Überschrift *Elektronen*, welches mit einem grünen Pfeil markiert ist. Die dargestellten Graphen sind gut erkennbar, da sie besser vom Hintergrund abgegrenzt werden

können als die schwarzen Überschriften auf den weißen Postern. Auch auf dem Poster rechts daneben wirken Abbildungen durch die gute Abgrenzung der verschiedenen Farben schärfer als ihre Überschriften. Der Patient soll die Umgebung in der Realität wiedererkennen können, dafür reichen begrenzte Immersion durch die teilweise fehlerhafte Zusammensetzung der Textur und partiell niedrige Auflösung aus.

### 5.2.5 Reduzierung der Faces

In einem Virtual Environment ist es wichtig einen immersiven Eindruck zu erwecken, dazu ist die Qualität der Technik und Modelle ausschlaggebend (siehe Abschnitt 2.3.3). Daher musste ein Kompromiss zwischen Qualität und Feinheit der Modelle sowie der technischen Leistungsfähigkeit des Systems gefunden werden. Um ein flüssiges VR-Erlebnis schaffen zu können, unterstützt das HTC Vive HMD bis zu 90 FPS. Damit auch bei Kopfdrehungen Animationen und Bewegungen flüssig wahrgenommen werden, sollten mindestens 70 FPS dauerhaft erreicht werden können. Wie in Abschnitt 4.2.7 beschrieben, sollte eine Schwelle von rund 200.000 Faces pro Modell eingehalten werden. Dies ist mit einer Textur der Größe 8192x8192 ausreichend, um ein Objekt der Größe 3x3x3m<sup>3</sup> ohne eine sichtbare Abrundung von rechtwinkligen Kanten zu erzeugen. Es wird empfohlen, die Reduzierung erst durchzuführen, nachdem eine Rekonstruktion ohne diese Reduzierung zu viele Faces als Ergebnis liefert.

## 5.3 Erfahrungen aus dem Aufbau des HTC Vive VR-Systems

Das HTC Vive VR-System gehört durch externes Tracking des Head-Mounted Displays sowie Nutzung der Ressourcen eines leistungsfähigen Computers für die Berechnung des Virtual Environment zu den professionellsten VR-Systemen. Im Gegensatz zu Head-Mounted Displays, in welche ein Smartphone eingesetzt wird und komplett ohne externes Tracking auskommen, müssen einige Grundvoraussetzungen geschaffen werden. Das System ist durch den Aufbau des Computers und der Stative mit den Base Stations nur begrenzt mobil, sodass eine Demonstration einen Grundaufwand zur Folge hat. Der optische Synchronisationsmodus erfordert eine genaue Ausrichtung der Base Stations, was für einen mobilen Aufbau nur eingeschränkt gewährleistet werden kann. Daher sollte der kabelbasierte Tracking-Modus verwendet werden. Zudem ist der kabelbasierte Modus weniger störanfällig. Ein weiterer Vorteil des kabelbasierten Modus ist, dass keine Sichtverbindung der beiden Kameras notwendig ist. Es genügt eine Sichtverbindung von mindestens einer Base Station zu den Controllern und dem Head-Mounted-Display. Der vorgestellte Aufbau bildet das Minimum an Raum, der für eine Demonstration notwendig wird und bildet daher einen Kompromiss zwischen Tracking-Genaugkeit und Raumverfügbarkeit. Sollte mehr Raum verfügbar sein, sollte der Aufbau vergrößert werden, um Tracking-Ungenauigkeiten zu reduzieren. Die Auflösung des Head-Mounted Displays ist für den Anwendungsfall ausreichend.

## 5.4 Entwicklung des Demonstrators mit der Unreal Engine

Dieser Abschnitt behandelt die Umsetzung des Demonstrators mit der Unreal Engine. Zunächst wird das Konzept hinter der Entwicklung und anschließend dessen Umsetzung und dabei aufgetretene Probleme diskutiert.

### 5.4.1 Datenmodell und Gruppierung von Aktoren

Das Datenmodell des Demonstrators wächst mit einer Integration weiterer animierter Objekte. Damit die Animationen unter den Objekten abgestimmt sind, sollten diese in Gruppen eingeteilt werden und gruppenweise parametrisiert werden. Das ist notwendig, damit eine gemeinsame Parametrisierung der Aktoren möglich ist. Jede Gruppe besitzt dabei ihr eigenes Datenmodell. Im Demonstrator wurde eine Einteilung von Linearbeschleuniger, Positionierungslaser Bestrahlungsstrahl und Gantry in drei Gruppen vorgenommen. Alle genannten Objekte werden gemeinsam parametrisiert, wobei die Positionierungslaser und der Bestrahlungsstrahl mit der Gantry eine weitere Untergruppe bilden, sowie die Patientenliege alleine (vgl. Abbildung 4.12 - alle Klassen mit *Configuration* im Namen). Es muss beachtet werden, dass Objekte nur gemeinsam parametrisiert werden sollten, sofern diese auch in einem miteinander agierenden Verhältnis stehen. Die Gruppierung ermöglicht eine Wiederverwendung der Gruppen in mehreren Szenen mit verschiedenen Parametrisierungen, ohne für jede Szene ein Datenmodell definieren zu müssen.

### 5.4.2 Beleuchtung

Die Beleuchtung der Szene ist essentiell um einen realen Eindruck zu erwecken. Dabei sollte nicht nur auf eine gleichmäßige Beleuchtung der Szene geachtet werden, sondern auch darauf, dass Modelle akzentuiert werden, die besonders relevant für das zu erklärende Konzept sind. Dies kann z.B. mit einem Spotlight erreicht werden, das das Modell direkt anleuchtet. Dabei müssen die Objekte gezielt ausgewählt werden, da jedes zusätzliche Licht in der Szene Rechenleistung durch z.B. Schattenwurf und Reflexion benötigt. Dies ist nicht zu vernachlässigen, da die komplexen Texturen der Modelle und auch das statische Umgebungslicht mit sogenannten *Light Maps* vorberechnet werden. Jede Veränderung der Gegebenheiten, wie z.B. eine Animation eines komplexen Meshes mit Textur, welches durch ein Spotlight angeleuchtet wird, erfordert, dass die Beleuchtung während der Simulation neu berechnet wird.

### 5.4.3 Kollimation und Darstellung von Strahlenform

Der Bestrahlungsstrahl wurde durch ein Partikelsystem realisiert (siehe Abschnitt 4.5.9). Die Simulation des Strahls weicht dabei durch die Umsetzung von der Realität

ab. Der Strahl ist nicht formbar und kann nicht durch Blenden beschränkt werden, sondern verhält sich atomar. Es muss im realen Anwendungsfall geprüft werden, ob es notwendig ist, eine physikalisch korrekte Simulation des Strahls zu realisieren.

### 5.4.4 Darstellung der Positionierungslaser

Die Darstellung der Laser erfolgt durch ein Partikelsystem, wie es in Abbildung 4.25 für den Bestrahlungsstrahl dargestellt ist. Die Laser wirken real, was durch ihren Fokus und das Leuchten unterstützt wird. In der Realität sind die Positionierungslaser Ebenen, die aus den drei Raumrichtungen die Achsen des Koordinatensystems mit dem Isozentrum als Ursprung markieren. Mit der Unreal Engine besitzen Beam-Partikelsysteme zwar eine Breite, können jedoch nicht als Ebene dargestellt werden. Damit die Positionierungslaser als Ebene dargestellt werden können, muss das Partikelsystem durch ein Mesh ersetzt werden, wie für den Bestrahlungsstrahl in Abschnitt 5.4.3 vorgestellt wurde. Dabei treten die selben Probleme der Lücke und Überlappung des Meshs auf der Oberfläche eines Objekts auf. Eine Aufteilung der Ebene und Approximation repräsentiert dabei ein Linien- statt ein Oberflächenintegral. In der Simulation ist dies ein Anwendungsfall, an dem man bewusst von der Realität abweichen kann. Denn eine Positionierung kann auch mit einfachen Lasern erfolgen. Durch das Prinzip des *Willentlichen Ausblendens des Unglaubens* ist es wichtig, dass ein Laser vorhanden ist, die Form ist irrelevant, da sie den Patienten nicht direkt betrifft.

## 5.5 Ausblick

Es bleibt abzuwarten, ob die Technologie den Sprung in die Medizin schafft. Um diese in der Medizin sinnvoll verwenden zu können, müssen eine Zielgruppe und ein Anwendungsgebiet identifiziert werden, in welchem sie einen Mehrwert zu bisherigen Methoden erbringen kann. In der klinischen Routine werden viele Daten erhoben, die primär vom medizinischen Personal für die medizinische Versorgung verwendet werden. Nach kurzer Darstellung gegenüber dem Patienten ist meist nicht vorgesehen, die Daten erneut in Kombination mit einer Neuerfassung zu präsentieren, da der meist kurze Zeitraum von Diagnostik, Therapie und Genesung dies in der Regel nicht rechtfertigt. In der Strahlentherapie fallen eine Vielzahl von Daten während der Planung und Therapie an, die auch eine regelmäßige Darlegung eines Zwischenstands mit z.B. Virtual Reality ermöglichen.

Weiterhin können die Methodik und das Konzept der Information durch Simulation auch auf die Ausbildung und Schulung von medizinischem Personal angewandt werden. Dies schafft teilweise andere Anforderungen an die Korrektheit und Qualität der Simulationen, sodass eine vorhergehende Analyse dieses Anwendungsfalls notwendig ist.

## *5 Diskussion*

---

Der Effekt und die Akzeptanz der Technik spielen eine wichtige Rolle für den Einsatz der Methodik. Diese müssen durch eine Evaluation im klinischen Alltag evaluiert werden, wofür die Entwicklung einer weitergehenden Software auf Basis der gewonnenen Erkenntnisse notwendig wird.

## Literatur

- [Ala11] Alaraj A, Lemole MG, Finkle JH et al. „Virtual reality training in neurosurgery: Review of current status and future applications“. *Surg Neurol Int* 2 (2011). DOI: 10.4103/2152-7806.80117.
- [And84] Andersen BL, Karlsson JA, Anderson B et al. „Anxiety and Cancer Treatment: Response to Stressful Radiotherapy“. *Health psychology : official journal of the Division of Health Psychology, American Psychological Association* 3.6 (1984), S. 535–551.
- [And85] Andersen BL und Tewfik HH. „Psychological Reactions to Radiation Therapy: Reconsideration of the Adaptive Aspects of Anxiety“. *Journal of personality and social psychology* 48.4 (1985), S. 1024–1032.
- [Ben03] Bennenbroek FTC, Buunk BP, Stiegelis HE et al. „Audiotaped social comparison information for cancer patients undergoing radiotherapy: differential effects of procedural, emotional and coping information.“ *Psycho-oncology* 12 (6 2003), S. 567–79. DOI: 10.1002/pon.674.
- [Bri90] Bricken W. „Virtual Reality: Directions of Growth“. *Notes from the SIGGRAPH '90 Panel (HITL Technical Report R-90-1)* (1990).
- [But97] Butow PN, Maclean M, Dunn SM et al. „The dynamics of change: cancer patients' preferences for information, involvement and support.“ *Annals of oncology : official journal of the European Society for Medical Oncology* 8 (9 1997), S. 857–63.
- [But98] Butow P, Brindle E, McConnell D et al. „Information booklets about cancer:“ *Patient Education and Counseling* 33.2 (1998), S. 129–141. DOI: 10.1016/S0738-3991(97)00071-2.
- [Cas80] Cassileth BR, Zupkis RV, Sutton-Smith K et al. „Information and participation preferences among cancer patients.“ *Annals of internal medicine* 92 (6 1980), S. 832–6.
- [Cig08] Cignoni P, Callieri M, Corsini M et al. „MeshLab: an Open-Source Mesh Processing Tool“. *Eurographics Italian Chapter Conference*. Hrsg. von Scariano V, Chiara RD und Erra U. The Eurographics Association, 2008. DOI: 10.2312/LocalChapterEvents/ItalChap/ItalianChapConf2008/129-136.
- [Clo99] Clotfelter CE. „The effect of an educational intervention on decreasing pain intensity in elderly people with cancer.“ *Oncology nursing forum* 26 (1 1999), S. 27–33.
- [Cry16] Crytek GmbH. *VR Support*. 2016. URL: <http://docs.cryengine.com/display/CEMANUAL/VR+Support> (Zugriff am 15. Okt. 2017).

- [Dör13] Dörner R. *Virtual und Augmented Reality (VR / AR) : Grundlagen und Methoden der Virtuellen und Augmentierten Realität*. Berlin, Heidelberg: Springer-Verlag GmbH Berlin Heidelberg, 2013. DOI: 10.1007/978-3-642-28903-3.
- [Dry95] Drysdale RLS, Rote G und Aichholzer O. *A simple linear time greedy triangulation algorithm for uniformly distributed points*. Institutes for Information Processing Graz, 1995.
- [Dun93] Dunn SM, Butow PN, Tattersall MH et al. „General information tapes inhibit recall of the cancer consultation.“ *Journal of Clinical Oncology* 11.11 (1993), S. 2279–2285. DOI: 10.1200/JCO.1993.11.11.2279.
- [EGP12] EGPJET3D. *Northrop T-38 Talon*. 2012. URL: <https://www.turbosquid.com/FullPreview/Index.cfm/ID/711530> (Zugriff am 5. Okt. 2017).
- [Els17] Elsner K, Naehrig D, Halkett GKB et al. „Reduced patient anxiety as a result of radiation therapist-led psychosocial support: a systematic review“. *Journal of Medical Radiation Sciences* (2017), n/a–n/a. DOI: 10.1002/jmrs.208.
- [Epi17] Epic Games, Inc. *Unreal Engine Features*. 2017. URL: <https://www.unrealengine.com/en-US/features> (Zugriff am 15. Okt. 2017).
- [Epia] Epic Games, Inc. *Unreal Engine Features*. URL: <https://www.unrealengine.com/unreal-engine-4> (Zugriff am 21. Apr. 2017).
- [Epib] Epic Games, Inc. *What Is Unreal Engine 4*. URL: <https://www.unrealengine.com/what-is-unreal-engine-4> (Zugriff am 21. Apr. 2017).
- [Fal95] Fallowfield L, Ford S und Lewis S. „No news is not good news: Information preferences of patients with cancer“. *Psycho-Oncology* 4.3 (1995), S. 197–202. DOI: 10.1002/pon.2960040305.
- [Fla10] Flavell L. *Beginning Blender: Open Source 3D Modeling, Animation, and Game Design*. Apresspod Series. Apress, 2010.
- [For95] Ford S, Fallowfield L, Hall A et al. „The influence of audiotapes on patient participation in the cancer consultation“. *European Journal of Cancer* 31.13 (1995), S. 2264–2269. DOI: 10.1016/0959-8049(95)00336-3.
- [Gai17a] Gaillard F. *Case courtesy of A.Prof Frank Gaillard, Radiopaedia.org, rID: 4090*. 2017. URL: <https://radiopaedia.org/play/2310/entry/28396/case/4090/studies/6567> (Zugriff am 17. Okt. 2017).
- [Gai17b] Gaillard F. *Case courtesy of A.Prof Frank Gaillard, Radiopaedia.org, rID: 4758*. 2017. URL: <https://radiopaedia.org/articles/intracranial-metastatic-melanoma-2> (Zugriff am 17. Okt. 2017).

- [Ger17] Gerlach S, Kuhlemann I, Ernst F et al. „Impact of robotic ultrasound image guidance on plan quality in SBRT of the prostate“. *The British Journal of Radiology* 90.1078 (2017), S. 20160926. DOI: 10.1259/bjr.20160926.
- [Gon13] González-Arriagada WA, Andrade MAC de, Ramos LMA et al. „Evaluation of an educational video to improve the understanding of radiotherapy side effects in head and neck cancer patients“. *Supportive Care in Cancer* 21.7 (2013), S. 2007–2015. DOI: 10.1007/s00520-013-1730-x.
- [Goo17a] Google Inc. *ARCore: Augmented reality at Android scale*. 2017. URL: <https://www.blog.google/products/google-vr/arcore-augmented-reality-android-scale/> (Zugriff am 5. Okt. 2017).
- [Goo17b] Google Inc. *ARCore Overview*. 2017. URL: <https://developers.google.com/ar/discover/> (Zugriff am 4. Okt. 2017).
- [Goo17c] Google Inc. *Tango Concepts*. 2017. URL: <https://developers.google.com/tango/overview/concepts> (Zugriff am 4. Okt. 2017).
- [Gut08] Gutiérrez A, Mario A, Thalmann D et al. *Stepping into Virtual Reality*. London: Springer London, 2008. DOI: 10.1007/978-1-84800-117-6.
- [Hac00] Hack TF, Pickles T, Bultz BD et al. „Feasibility of an Audiotape Intervention for Patients with Cancer“. *Journal of Psychosocial Oncology* 17.2 (2000), S. 1–15. DOI: 10.1300/j077v17n02\_01.
- [Häg01] Häggmark C, Bohman L, Ilmoni-Brandt K et al. „Effects of information supply on satisfaction with information and quality of life in cancer patients receiving curative radiation therapy“. *Patient Education and Counseling* 45.3 (2001), S. 173–179. DOI: 10.1016/S0738-3991(01)00116-1.
- [Hag96] Hagopian GA. „The effects of informational audiotapes on knowledge and self-care behaviors of patients undergoing radiation therapy“. *Oncol Nurs Forum* 23.4 (1996), S. 697–700.
- [Hah05] Hahn CA, Fish LJ, Dunn RH et al. „Prospective trial of a video educational tool for radiation oncology patients.“ *American journal of clinical oncology* 28 (6 2005), S. 609–12. DOI: 10.1016/j.ijrobp.2004.07.530.
- [Hal12] Halkett GK, Kristjanson LJ, Lobb E et al. „Information needs and preferences of women as they proceed through radiotherapy for breast cancer“. *Patient Education and Counseling* 86.3 (2012), S. 396–404. DOI: 10.1016/j.pec.2011.05.010.
- [Hal09] Halkett GK, Short M und Kristjanson LJ. „How do radiation oncology health professionals inform breast cancer patients about the medical and technical aspects of their treatment?“ *Radiotherapy and Oncology* 90.1 (2009), S. 153–159. DOI: 10.1016/j.radonc.2008.05.017.

- [Har00] Harrison R, Dey P, Slevin NJ et al. „Randomized controlled trial to assess the effectiveness of a videotape about radiotherapy“. *British Journal of Cancer* 84.1 (2000), S. 8–10.
- [Har93] Harrison-Woermke DE und Graydon JE. „Perceived informational needs of breast cancer patients receiving radiation therapy after excisional biopsy and axillary node dissection“. *Cancer Nursing* 16.6 (1993).
- [Hin95] Hinds C, Streater A und Mood D. „Functions and preferred methods of receiving information related to radiotherapy. Perceptions of patients with cancer.“ *Cancer nursing* 18 (5 1995), S. 374–84.
- [Hug15] Hughes M, Sykes V, Jessop A et al. „PO-1126 A pilot study using a computer simulation tool for patient/carer education of radiotherapy“. *Radiotherapy and Oncology* 115 (2015), S611–S612. DOI: 10.1016/S0167-8140(15)41118-1.
- [RTAB] IntRoLab. *RTAB-Map: Real-Time Appearance-Based Mapping*. URL: <http://introlab.github.io/rtabmap/> (Zugriff am 20. Sep. 2017).
- [Kah17] Kahl-Scholz M und Vockelmann C, Hrsg. *Basiswissen Radiologie. Nuklearmedizin und Strahlentherapie*. Berlin, Heidelberg: Springer-Verlag GmbH, 2017. DOI: 10.1007/978-3-662-54278-1.
- [Kaz06] Kazhdan M, Bolitho M und Hoppe H. „Poisson Surface Reconstruction“. *Proceedings of the Fourth Eurographics Symposium on Geometry Processing*. SGP '06. Cagliari, Sardinien, Italien: Eurographics Association, 2006, S. 61–70.
- [Kaz07] Kazhdan M, Klein A, Dalal K et al. „Unconstrained Isosurface Extraction on Arbitrary Octrees“. *Proceedings of the Fifth Eurographics Symposium on Geometry Processing*. SGP '07. Barcelona, Spanien: Eurographics Association, 2007, S. 125–133.
- [Kho16] Khor WS, Baker B, Amin K et al. „Augmented and virtual reality in surgery—the digital surgical environment: applications, limitations and legal pitfalls“. *Ann Transl Med* 4.23 (2016). DOI: 10.21037/atm.2016.12.23.
- [Kil15] Kilgus T, Heim E, Haase S et al. „Mobile markerless augmented reality and its application in forensic medicine“. *International Journal of Computer Assisted Radiology and Surgery* 10.5 (2015), S. 573–586. DOI: 10.1007/s11548-014-1106-9.
- [Kno02] Knox R, Butow PN, Devine R et al. „Audiotapes of oncology consultations: only for the first consultation?“ *Annals of Oncology* 13.4 (2002), S. 622–627. DOI: 10.1093/annonc/mdf055.

- [Lab13] Labb   M und Michaud F. „Appearance-Based Loop Closure Detection for Online Large-Scale and Long-Term Operation“. *IEEE Transactions on Robotics* 29.3 (2013), S. 734–745. DOI: 10.1109/TRO.2013.2242375.
- [Lab14] Labb   M und Michaud F. „Online global loop closure detection for large-scale multi-session graph-based SLAM“. *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2014, S. 2661–2666. DOI: 10.1109/IROS.2014.6942926.
- [Lam94] Lamszus K, Verres R und Hubener KH. „[How do patients experience radiotherapy?].“ *Strahlentherapie und Onkologie : Organ der Deutschen Rontgengesellschaft ... [et al]* 170 (3 1994), S. 162–8.
- [Len17] Lenovo. *Phab 2 Pro*. 2017. URL: <https://www3.lenovo.com/de/de/smарт-devices/lenovo/phab-series/Lenovo-PB2-690M/p/ZZITZTPPB4M> (Zugriff am 4. Okt. 2017).
- [Lil14] Lile D. *Blender Character Modeling 1 of 10*. 2014. URL: <https://www.youtube.com/watch?v=0QT1GNMevfc&t=583s> (Zugriff am 9. Okt. 2017).
- [Lon01] Long LE. „Being Informed: Undergoing Radiation Therapy“. *Cancer Nursing* 24.6 (2001).
- [McD01] McDerby MJ, John NW, Brunt JN et al. „Modelling concepts of proton eye radiotherapy“. *Physiol Meas* 22.3 (2001), S. 611–23.
- [McH95] McHugh P, Lewis S, Ford S et al. „The efficacy of audiotapes in promoting psychological well-being in cancer patients: a randomised, controlled trial.“ *British Journal of Cancer* 71.2 (1995), S. 388.
- [Med16] Medienzentrum Universit  tsklinikum Heidelberg. *Protonentherapie und Schwerionentherapie – hochpr  zise Behandlung gegen Krebs*. 2016. URL: <https://www.heidelberg-university-hospital.com/de/erkrankungen-behandlungen/tumorerkrankungen/protonen-und-schwerionentherapie/> (Zugriff am 6. Okt. 2017).
- [Mer96] Meredith C, Symonds P, Webster L et al. „Information needs of cancer patients in west Scotland: cross sectional survey of patients' views.“ *BMJ (Clinical research ed.)* 313 (7059 1996), S. 724–6. DOI: 10.1136/bmj.313.7059.724.
- [Mes01] Mesters I, Borne B van den, De Boer M et al. „Measuring information needs among cancer patients.“ *Patient education and counseling* 43 (3 2001), S. 253–62. DOI: 10.1016/S0738-3991(00)00166-X.
- [Moo88] Mood DW, Cook CA und Chadwell DK. „Increasing patients' knowledge of radiation therapy“. *Int J Radiat Oncol Biol Phys* 15.4 (1988), S. 989–93.

- [Nic15] Nickel F, Brzoska JA, Gondan M et al. „Virtual Reality Training Versus Blended Learning of Laparoscopic Cholecystectomy: A Randomized Controlled Trial With Laparoscopic Novices“. *Medicine (Baltimore)* 94.20 (2015). DOI: 10.1097/md.0000000000000764.
- [Nor92] North N, Cornbleet MA, Knowles G et al. „Information giving in oncology: A preliminary study of tape-recorder use“. *British Journal of Clinical Psychology* 31.3 (1992), S. 357–359. DOI: 10.1111/j.2044-8260.1992.tb01005.x.
- [Oya97] Oyama H, Wakao F und Okamura H. „Virtual reality support system in palliative medicine“. *Stud Health Technol Inform* 39 (1997), S. 60–3.
- [Rai85] Rainey LC. „Effects of preparatory patient education for radiation oncology patients“. *Cancer* 56.5 (1985), S. 1056–1061. DOI: 10.1002/1097-0142(19850901)56:5<1056::AID-CNCR2820560516>3.0.CO;2-Z.
- [Rey78] Reynolds M. „No news is bad news: patients’ views about communication in hospital.“ *British Medical Journal* 1.6128 (1978), S. 1673–1676.
- [Rey81] Reynolds PM, Sanson-Fisher RW, Poole AD et al. „Cancer and communication: information-giving in an oncology clinic.“ *British Medical Journal (Clinical research ed.)* 282.6274 (1981), S. 1449–1451.
- [Rot95] Rothbaum B, Hodges L, Kooper R et al. „Effectiveness of computer-generated (virtual reality) graded exposure in the treatment of acrophobia“. *American Journal of Psychiatry* 152.4 (1995), S. 626–628. DOI: 10.1176/ajp.152.4.626.
- [Rut05] Rutten LJF, Arora NK, Bakos AD et al. „Information needs and sources of information among cancer patients: a systematic review of research (1980–2003)“. *Patient Education and Counseling* 57.3 (2005), S. 250–261. DOI: 10.1016/j.pec.2004.06.006.
- [Sch12] Scheffel H, Alkadhi H, Boss A et al. *Praxisbuch MRT Abdomen und Becken*. SpringerLink : Bücher. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, Online-Ressource (250S. 414 Abb., 1 Abb. in Farbe, digital). DOI: 10.1007/978-3-642-25340-9.
- [Sch03] Schneider SM, Ellis M, Coombs WT et al. „Virtual Reality Intervention for Older Women with Breast Cancer“. *Cyberpsychol Behav* 6.3 (2003), S. 301–7. DOI: 10.1089/109493103322011605.
- [Sch07] Schneider SM und Hood LE. „Virtual Reality: A Distraction Intervention for Chemotherapy“. *Oncol Nurs Forum* 34.1 (2007), S. 39–46. DOI: 10.1188/07.onf.39–46.

- [Shr12] Shrimpton BJM, Willis DJ, Tongs CD et al. „Movie making as a cognitive distraction for paediatric patients receiving radiotherapy treatment: qualitative interview study“. *BMJ Open* 3.1 (2012), e001666. DOI: 10.1136/bmjopen-2012-001666.
- [Sla09] Slater M. „Place illusion and plausibility can lead to realistic behaviour in immersive virtual environments.“ *Philosophical transactions of the Royal Society of London. Series B, Biological sciences* 364 1535 (2009), S. 3549–57.
- [Sla97] Slater M und Wilbur S. „A Framework for Immersive Virtual Environments (FIVE): Speculations on the Role of Presence in Virtual Environments“. *Presence: Teleoperators and Virtual Environments* 6.6 (1997), S. 603–616. DOI: 10.1162/pres.1997.6.6.603.
- [Smi16] Smith S, Yan B, Milross C et al. „Radiation therapy for people with cancer: what do written information materials tell them?“ *European Journal of Cancer Care* 25.4 (2016), S. 675–685. DOI: 10.1111/ecc.12366.
- [Sti17] Stichting Blender Foundation. *The software*. 2017. URL: <https://www.blender.org/about/> (Zugriff am 5. Okt. 2017).
- [Sut89] Sutherland HJ, Llewellyn-Thomas HA, Lockwood GA et al. „Cancer patients: their desire for information and participation in treatment decisions.“ *Journal of the Royal Society of Medicine* 82.5 (1989), S. 260–263.
- [Sut65] Sutherland IE. „The Ultimate Display“. *Proceedings of the IFIP Congress*. 1965, S. 506–508.
- [Tan17] Tang R, Ma L, Xiang C et al. „Augmented reality navigation in open surgery for hilar cholangiocarcinoma resection with hemihepatectomy using video-based in situ three-dimensional anatomical modeling: A case report“. *Medicine* 96.37 (2017), e8083. DOI: 10.1097/MD.0000000000008083.
- [Tho00] Thomas R, Daly M, Perryman B et al. „Forewarned is forearmed—benefits of preparatory information on video cassette for patients receiving chemotherapy or radiotherapy—a randomised controlled trial.“ *European journal of cancer (Oxford, England : 1990)* 36 (12 2000), S. 1536–43. DOI: 10.1016/S0959-8049(00)00136-2.
- [Tho99a] Thomas R, Thornton H und Mackay J. „Patient Information Materials in Oncology: Are they Needed and do they Work?“ *Clinical Oncology* 11.4 (1999), S. 225–231. DOI: 10.1053/clon.1999.9054.
- [Tho99b] Thomas, Deary, Kaminski et al. „Patients’ preferences for video cassette recorded information: effect of age, sex and ethnic group“. *European Journal of Cancer Care* 8.2 (1999), S. 83–86. DOI: 10.1046/j.1365-2354.1999.00123.x.

- [Tim05] Timmermans LM, Maazen RW van der, Verhaak CM et al. „Patient participation in discussing palliative radiotherapy“. *Patient Educ Couns* 57.1 (2005), S. 53–61. DOI: 10.1016/j.pec.2004.03.016.
- [Tri00] Triggs B, McLauchlan P, Hartley R et al. „Bundle Adjustment – A Modern Synthesis“. *International Workshop on Vision Algorithms*. Bd. 1883. Corfu, Greece: Springer-Verlag, 2000, S. 298–372. DOI: 10.1007/3-540-44480-7\_21.
- [Uni17] Unity Technologies. *Unity 2017: Die weltweit führende Creation-Engine für Gaming*. 2017. URL: <https://unity3d.com/de/unity> (Zugriff am 15. Okt. 2017).
- [Uni12] Universitätsklinikum Heidelberg. *Nach dem Synchrotron teilt sich die Strahlführung auf in Richtung der drei Behandlungsplätze. Das HIT verfügt über zwei horizontale Behandlungsplätze und einen Gantry-Behandlungsplatz. Zusätzlich gibt es einen Bestrahlungsplatz speziell für die Grundlagenforschung*. 2012. URL: [https://www.klinikum.uni-heidelberg.de/fileadmin/pressestelle/Fotostrecke\\_GantryEinweihung/05\\_HIT\\_Strahlfuehrung.jpg](https://www.klinikum.uni-heidelberg.de/fileadmin/pressestelle/Fotostrecke_GantryEinweihung/05_HIT_Strahlfuehrung.jpg) (Zugriff am 6. Okt. 2017).
- [Val16] Valve Corporation. *HTC Vive Installation Guide*. 2016. URL: [https://support.steampowered.com/steamvr/HTC\\_Vive/](https://support.steampowered.com/steamvr/HTC_Vive/) (Zugriff am 8. Aug. 2017).
- [Wan13] Wannenmacher M, Wenz F und Debus J. *Strahlentherapie*. 2. Aufl. Berlin, Heidelberg: Springer-Verlag GmbH, 2013. DOI: 10.1007/978-3-540-88305-0.
- [Wil17] Williams K, Blencowe J, Ind M et al. „Meeting radiation therapy patients informational needs through educational videos augmented by 3D visualisation software“. *Journal of Medical Radiation Sciences* 64.1 (2017), S. 35–40. DOI: 10.1002/jmrs.220.
- [Zeg12] Zeguers M, Haes HC de, Zandbelt LC et al. „The Information Needs of New Radiotherapy Patients: How to Measure? Do They Want to Know Everything? And if Not, Why?“ *International Journal of Radiation Oncology\*Biology\*Physics* 82.1 (2012), S. 418–424. DOI: 10.1016/j.ijrobp.2010.09.032.

## **Danksagung**

An dieser Stelle möchte ich mich bei meiner Betreuerin Frau Dr. Kristina Giske für die engagierte Betreuung und die vielen hilfreichen Gespräche bedanken.

Ein weiterer Dank geht an Herrn Prof. Dr. Bendl für das Feedback und die Motivation während der Bearbeitung des Themas.

Für den häufig spontanen Zugang zum Linearbeschleuniger möchte ich mich bei den Mitgliedern der Arbeitsgruppe E0407 bedanken.

Für die Bereitstellung und Suche nach bestehenden 3D-Modellen bedanke ich mich bei Herrn Gernot Echner.

Dank gilt weiterhin allen Personen, die mich bei dieser Arbeit unterstützt oder während des Studiums begleitet haben.



## **Eidesstattliche Erklärung**

Ich erkläre hiermit an Eides statt, dass ich die vorliegende Arbeit selbstständig und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe; die aus fremden Quellen (einschließlich elektronischer Quellen) direkt oder indirekt übernommenen Gedanken sind als solche kenntlich gemacht.

Heidelberg, 2. November 2017

---

(Ort, Datum)

(Unterschrift)