

Otto-von-Guericke-Universität Magdeburg

Fakultät für Informatik



Projektbericht
Wissenschaftliches Teamprojekt

Exploration medizinischer Bild Datensätze mit der HTC Vive

Autoren:

Sebastian Heerwald, Franz Kuntke

09.07.2017

Betreuer:

Prof. Dr.-Ing. habil. Bernhard Preim

Institut für Simulation und Graphik

M.Sc. Patrick Saalfeld

Institut für Simulation und Graphik

Abstract

With the introduction of the last generation of head-mounted displays it became less expensive and easier to obtain high immersive virtual reality hardware. As a result more areas of applications start to exploit the possibilities of this technology. Previous scientific work discussed advantages of immersive systems in medical contexts, but a direct comparison between a last generation head-mounted display and a common 2D system for medical images is missed to date.

In this scientific teamproject, we created such a comparison between a virtual reality software and a 2D screen software, both in context of medical image exploration. Therefore we carried out a comparative user study between a self developed virtual reality software for medical image exploration and a commercial tool for medical image analysis for common 2D screens.

The design and development of our virtual reality software prototype was driven by the possibilities of the HTC Vive and the associated controllers. This allowed us to create an easy-to-use software with basic interactions – rotation, translation and scaling – and tools – ruler and clipping plane. The test persons were asked to measure the size of given structures in both systems. As a result, the comparative user study shows, that the time needed for the tasks was in average lower when using our virtual reality software prototype. Surprisingly the resulting measurement values differ remarkably between both used systems. Evaluating the test persons' opinion via questionnaire, we also show, that both systems seem to require similar mental effort by a user, although the physical effort is higher when using the virtual reality system. Additionally most test persons think that the measurement results are more accurate when using the virtual reality system.

1. Einführung

Virtual Reality (VR-) Systeme haben in den letzten Jahren große technologische Fortschritte erfahren. Zu den wesentlichen Verbesserungen zählt die erhöhte Bildqualität der Head Mounted Displays (HMDs), die unter Anderem durch eine vergrößerte Auflösung von 640 x 480 Pixel pro Auge auf 1080 x 1200 Pixel pro Auge zu Stande kommt. Weiterhin haben sich in die Systeme integrierte Positionserkennung von HMD (Kopfposition) und Controllern (Handposition) etabliert. Durch sinkende Preise der VR-Headsets und entsprechender Eingabegeräte erhöht sich die Massentauglichkeit der Hardware und somit erschließen sich neue Zielgruppen für VR-Software. Nun liegt es an der Software, von den neuen Möglichkeiten der Interaktion zu profitieren und passende Bedienkonzepte bereitzustellen, die einen Mehrwert erzeugen. In dieser Arbeit wird evaluiert, welche Vorteile eine VR-Umgebung bezüglich der Exploration medizinischer Datensätze mit sich bringt. Dazu wurde ein Softwareprototyp für die HTC Vive entwickelt, der Basisinteraktionen wie Rotation, Translation, Skalierung und Werkzeuge zum Messen anbietet. Der entwickelte Prototyp wurde im Rahmen einer komparativen Nutzerstudie mit einer kommerziellen Segmentierungs-, Visualisierungs- und Analysesoftware für medizinische Daten verglichen. Dieser Vergleich umfasst sowohl eine informelle Auswertung bezüglich der empfundenen Genauigkeit und Ermüdungserscheinungen, als auch eine formale Analyse der benötigten Zeit pro Aufgabe und System.

2. Verwandte Arbeiten

Das wissenschaftliche Teamprojekt hat starken Bezug zu drei Forschungsfeldern:

- (1) Untersuchung der Perzeption unterschiedlicher Technologien zur Darstellung von virtuellen Umgebungen, die über herkömmliche 2D-Monitor-Setups hinausgehen.
- (2) Entwicklung und Evaluation moderner Eingabekonzepte für virtuelle Umgebungen unter Einbeziehung von Eingabegeräten, die ohne Auflagefläche (engl. *support surface*) auskommen.
- (3) Interaktionen mit medizinischen Datensätzen, z. B. das Vermessen von dreidimensionalen Strukturen.

Im Folgenden werden Arbeiten aus den genannten Forschungsfeldern vorgestellt.

2.1 Perzeption durch Ausgabe-Technologien

Stereoskopische Displays stellen eine Weiterentwicklung herkömmlicher 2D-Monitore dar. Durch die Darstellung leicht unterschiedlicher Bilder für das linke und rechte Auge wird eine räumliche Darstellung ermöglicht. Oft wird für die stereoskopische Darstellung auf Filter in Form einer Brille gesetzt, um die beiden unterschiedlichen Bilder jeweils nur auf das gewünschte Auge zu projizieren. Bei auto-stereoskopischen Displays werden die darzustellenden Bilder bereits am Display mit separaten Winkeln ausgesandt, so dass die Bilder jeweils nur auf das gewünschte Auge treffen. Dadurch benötigt der Nutzer für die räumliche Wahrnehmung keine Brille. In der Arbeit von Baer et al. [BHS⁺14] wird durch eine Nutzerstudie die Anwendung im medizinischen Kontext eines solchen auto-stereoskopischen Displays mit 2D-Monitoren verglichen. Die Nutzer zeigen dabei eine klare Präferenz des 3D-Displays gegenüber des 2D-Displays bezüglich der Wahrnehmung der dargestellten 3D-Szene. Auch die Positionierung von virtuellen Objekten innerhalb der 3D-Szene wurde mit dem 3D-Display besser durchgeführt.

Darüber hinaus gibt es stereoskopische Displays mit Headtracking, beispielsweise von der Firma zSpace¹. Die zusätzliche Information über die Position des Kopfes ermöglicht es dabei, die Perspektive der virtuellen Kamera innerhalb der dargestellten Szene entsprechend der Bewegung des Nutzers anzupassen. Dies ermöglicht es mittels Bewegung des Kopfes, Objekte von unterschiedlichen Blickwinkeln zu betrachten. Durch die begrenzte Bildschirmfläche ist es allerdings beispielsweise nicht möglich, nur durch Bewegen des Kopfes, die Szene aus allen Blickrichtungen zu erfassen. Ein weiterer Nachteil der begrenzten Bildschirmfläche ist das eingeschränkte Sichtfeld, wodurch Objekte, die über den Rand des Sichtbereiches herausragen, abgeschnitten werden.

Eine höherer Grad an Immersion im Vergleich zu Displays ermöglichen Ausgabesysteme, die das gesamte Blickfeld umfassen. Bereits in den späten 60ern hat Ivan Sutherland erste Versuche mit Head-Mounted-Displays (HMDs) durchgeführt [Sut68]. Im Vergleich zu Monitoren wird die Umgebung, z. B. andere Menschen, die sich im Raum bewegen, ausgeblendet, wodurch sich der mentale Fokus komplett

¹zSpace, <https://zspace.com>, zuletzt abgerufen: 06.06.2017

auf die Szene legen kann. Die Pionierarbeit von Sutherland zeigte, dass Nutzer den räumlichen Eindruck und den hohen Grad an Realismus sehr positiv auffassen. Vor dem qualitativen Durchbruch der aktuellen HMDs, die seit Anfang 2016 für Konsumenten erhältlich sind, wurde für professionelle, immersive Umgebungen oft auf CAVEs (Cave Automatic Virtual Environments) [CNSD93] gesetzt. Ein CAVE ist ein begehbare Quader, dessen Flächen stereoskopische Darstellungen mittels Projektoren wiedergeben. Durch Headtracking in Kombination mit der stereoskopischen Darstellung kann dabei ein hohes Maß an Immersion erzielt werden. Es ergeben sich größere Anforderungen an die Entwicklung von VR-Umgebungen ergeben als bei herkömmlichen 2D-Monitor-Ausgaben: Um Kinetose (engl. *Motion Sickness*) zu vermeiden und trotzdem eine große Auflösung für die Displays bereitzustellen, ist es wichtig, dass die Software als auch Hardware eine hohe Berechnungsgeschwindigkeit zur Verfügung stellt. Erst dadurch wird eine interaktive und immersive Umgebung gewährleistet. Wird die Latenz zu hoch, ist das System für den Anwender nicht mehr angenehm benutzbar.

Mirhosseini et al. [MSG⁺14] stellen in ihrer Arbeit eine Übersicht von Vorteilen immersiver 3D Technologien für virtuelle Koloskopie auf. Die Autoren heben die verbesserte Wahrnehmung in immersiven Umgebungen hervor und erwarten Geschwindigkeitsvorteile im manuellen Aufspüren von Geschwülsten bei der virtuellen Koloskopie, begründet durch die Möglichkeit Oberflächengeometrien schneller zu erkennen.

Randall et al. [RMB⁺15] beschreiben eine konkrete Implementierung von virtueller Koloskopie unter Einbeziehung von HMDs – in ihrer Arbeit in Form eines Oculus Rift Development Kit 2. Drei Mediziner, die das entwickelte System getestet haben, bestätigen dabei, dass die intuitive Nutzung von Kopfbewegungen für das Anpassen der Ansicht in Kombination mit dem großen Sichtfeld und der stereoskopischen Darstellung Vorteile gegenüber herkömmlichen 2D-Monitor-Systemen hat. Als unkomfortabel empfinden die Nutzer dabei die Notwendigkeit, den Kopf um bis zu 180° drehen zu müssen. Die Autoren bescheinigen VR für virtuelle Koloskopie großes Potential, auch wenn weitere Forschung in diesem Bereich notwendig sei, unter anderem in Form von quantitativen Vergleichsstudien zwischen VR-Systemen und 2D-Monitor-Systemen.

In der Nutzerstudie von Venson et al. [VBM⁺16] wurde die Nutzbarkeit von VR für die radiologische Diagnose überprüft. In der Studie hatten 17 Mediziner die Aufgabe in zwei Volumendatensätzen Frakturen aufzufinden. Die Autoren kommen zum Ergebnis, dass ein VR-System eine hohe Effizienz im Auffinden von Frakturen hat. Ein Vergleich mit einem üblichen Ablauf an einem 2D-Monitor-System wurde von den Autoren leider nicht durchgeführt.

2.2 Eingabekonzepte für Virtuelle Umgebungen

Bewegen und manipulieren von 3D-Szenen ist mit herkömmlicher Maus-Tastatur Kombination nur mit vielen Interaktionen möglich. Eingaben sind somit mit einem hohen Interaktionsaufwand verbunden. Für die Interaktion in 3D-Szenen wurden deshalb neue Eingabegeräte entwickelt, welche mindestens 6 Freiheitsgrade (engl. *Degree-Of-Freedom*, *DOF*) besitzen, wie beispielsweise die Nintendo Wiimote oder

die HTC Vive Controller. Diese ermöglichen eine direktere Eingabe in 3D Umgebungen und haben den Vorteil, dass sie keine Auflagefläche benötigen, wodurch Aktionen auch fernab eines Schreibtisches ausgeführt werden können.

Gallo et al. [GDPM08] entwickelten für die Exploration medizinischer Volumendaten neue Interaktionskonzepte auf Basis der Nintendo Wiimote. Die Eingabe befindet sich dabei immer in einem von zwei möglichen Zuständen: Entweder dem *Pointing State* oder dem *Manipulation State*, wobei mittels Tastendruck zwischen beiden Modi gewechselt werden kann. Der Pointing State ermöglicht die Auswahl von Objekten im 3D Raum mit Hilfe eines Strahls, dessen Länge mittels zweier Tasten verkürzt oder verlängert werden kann. Im Manipulation State kann ein ausgewähltes Objekt gedreht, bewegt oder zugeschnitten werden. Für unsere Software setzen wir auf ein vergleichbares Konzept, so dass sich ein Controller zu jeder Zeit in einem State befindet, der über ein sichtbares Menü, welches in der immersiven Umgebung direkt an der Position der Controller gezeichnet wird, wechselbar ist. Die einzelnen States werden in Kapitel 4 beschrieben.

In der Arbeit von Beichel et al. [BS06] wird die Planung einer Leberoperation unter Nutzung einer VR-Umgebung untersucht. Die Eingabe in der VR-Umgebung wird dabei mit Hilfe eines eigens entwickelten 3D-Controller durchgeführt. Der Controller besitzt zwei Tasten, ein Scrollrad, sowie Marker für die räumliche Registrierung mittels Kameras. Die Kombination aus stereoskopischer Darstellung mit der 3D-Eingabe haben bei den Nutzern zu schnelleren Ergebnissen für die Planung der chirurgischen Eingriffe im Vergleich zu herkömmlichen Systemen mit 2D-Displays geführt.

2.3 Allgemeine Interaktionen mit medizinischen Volumendaten

Hänel et al. [HWHK16] untersuchen in ihrer Arbeit die Auswirkung unterschiedlicher Qualitätsparameter von medizinischen Volumenrendering und die damit verbundene Latenz in Bezug auf das Nutzerempfinden. Dabei wurde von Nutzern die kontinuierliche Anpassung der Rendering-Qualität als angenehm empfunden, bei der immer die beste Qualität, welche mit 60 Bildern pro Sekunde darstellbar ist angestrebt wird, es aber bis zu 2 Sekunden lang zu niedrigen Bildraten kommen kann. Im Vergleich dazu schnitten die Einstellungen sowohl mit allgemein höherer Darstellungsqualität bei geringerer Bildrate als auch mit niedrigerer Darstellungsqualität bei konstant über 60 Bildern pro Sekunde in der informellen Nutzerbefragung und der formalen Nutzeranalyse schlechter ab.

Typische Anwendungsszenarien bei der Evaluation medizinischer Volumendatensätze beinhalten das Ausmessen von Strukturen. Klassischerweise geschieht die visuelle Bemaßung über das Aufziehen einer Strecke, wobei die Länge der Strecke dem gesuchten Maß entspricht.

Rössling et al. [RCD⁺09] untersuchen die Akzeptanz von unterschiedlichen Darstellungen von Bemaßungen. Dabei wurden Bemaßungen in Form von Strecken mit unterschiedlichen Pfeilen (2D, 3D) als Endsegmente inkl. der textuellen Repräsentation des Messergebnisses direkt in der Szene (z.B. neben der Strecke oder innerhalb der Strecke als Unterbrechung) verglichen. Die Auswertung zeigte dabei, dass die Nutzer eine klare Zuordnung der Ergebniszahl zu der aufgezogenen Strecke, sowie

Kohärenz, d.h. das Vermeiden von sprunghaften Änderungen der Darstellung bei Änderung der Perspektive als angenehm empfinden. Dafür seien die Nutzer auch bereit Einschränkungen in Bezug auf die Lesbarkeit des Messwertes hinzunehmen. Im Resultat erscheint den Autoren für medizinische Visualisierung folgende Kombination als sinnvoll: 3D-Pfeile mit der Maßzahl als perspektivische Unterbrechung innerhalb der Strecke.

Um Daten möglichst genau zu messen, gibt es algorithmische Ansätze, welche segmentierte Strukturen mit unterschiedlichen Metriken, z.B. die kleinste und größte Ausdehnung einer Struktur, automatisch ausmessen. Dazu haben Rössling et al. in einer weiteren Arbeit [RCD⁺10] effiziente Algorithmen für die automatische Vermessung von Meshes entwickelt. Für eine medizinische Software zur Exploration und Analyse von volumetrischen Daten sind solche Ansätze sehr sinnvoll, da Nutzerfehler durch falsches Ausmessen vermieden werden können.

3. Konzept

Für den Vergleich der Exploration medizinischer Datensätze zwischen einer VR-Umgebung und einer klassischen Maus-Tastatur-Monitor Lösung wird jeweils eine entsprechende Software benötigt. Programme, die in die Kategorie einer klassischen Bedienung mit Maus und Tastatur fällt, gibt es reichlich. Für die modernen HMDs hingegen sind uns derzeit abseits wissenschaftlicher Prototypen keine Programme bekannt, die medizinische Bilddaten in einer VR-Umgebung visualisieren können und dazu einfache Explorationsaufgaben, beispielsweise das Messen von Strukturen, ermöglichen. Ein wesentlicher Bestandteil des wissenschaftlichen Teamprojektes ist es daher, eine Software zu entwickeln, welche die Durchführung der Nutzerstudie ermöglicht. Im folgenden werden Voraussetzungen seitens der zur Verfügung stehenden Hardware und Software aufgezeigt. Daraufhin stellen wir Anforderungen an die Implementierung auf und beschreiben im Anschluss mögliche Ideen und Konzepte, welche die Anforderungen erfüllen und somit für eine Umsetzung einer entsprechenden Software relevant sind.

3.1 Voraussetzungen durch Hardware

Für das wissenschaftliche Teamprojekt wurde die HTC Vive genutzt. Dieses VR-System besteht aus einem HMD, zwei Controllern für die Interaktion mit der Umwelt und zwei Lightroom-Boxen, die zum Tracking von Headset und Controller benötigt werden. Mithilfe der von den Lightroom-Boxen ausgesandten Laserimpulsen wird es dem HMD und den Controllern ermöglicht ihre absolute Position und Orientierung im Raum zu bestimmen. Damit dies funktioniert muss das System einmalig kalibriert werden. Anschließend verspricht es ein millimetergenaues Tracking. Position und Orientierung von HMD und Controllern, sowie Eingaben auf den Controllern können softwareseitig mit dem entsprechenden SDK ausgelesen werden.

Controller

Die Controller sind ergonomisch geformt. Sie werden vergleichbar mit einer Pistole in der Hand gehalten. Für die Interaktion sind die Controller mit verschiedenen Eingabeflächen und Buttons versehen. Ein Triggerbutton an der Rückseite des Controllers kann mithilfe des Zeigefingers gedrückt werden. Dieser Button ist als einziger Button analog, d. h. der Auslenkungswinkel ist bei Betätigung auslesbar. Zusätzlich besitzt der Triggerbutton einen spürbaren Einrastpunkt, ungefähr in der Mitte des drückbaren Weges. Dieser Einrastpunkt dient standardmäßig als Schwellwert, um zwischen den binären Zuständen *gedrückt* und *nicht gedrückt* zu unterscheiden. Auf der Oberseite des Controllers befindet sich ein kreisförmiges Touchpad, das gleichzeitig ein Button ist. Mithilfe dieses Touchpads ist es möglich direkt mit dem Daumen Interaktionen auszuführen, indem man darüber wischt oder eine Eingabe über die direkte Positionierung des Fingers auf dem Touchpad durchführt. Dadurch bietet der Controller eine frei belegbare Fläche, die beispielsweise mit mehreren virtuellen Buttons bestückt werden kann. Über und unter dem Touchpad befinden sich

kleine Menütasten, die für das Öffnen der Programm-Menüs genutzt werden können. An den unteren Rändern der Controller befinden sich Grip-Buttons, welche bei stärkerem Zugreifen am Controller ausgelöst werden.

Head-Mounted-Display

Das HMD besteht aus einem Display mit davor angebrachten Linsen. Das vom Display projizierte Bild erstreckt sich dabei durch die Verzerrung der Linsen über nahezu den gesamten Sichtbereich des Benutzers. Das Field-Of-View (FOV) beträgt dabei 110 Grad, was in etwa *üblichen* optischen Sehhilfen entsprechen. Da es sich um ein umschließendes Headset handelt sind Sichtbereiche, die nicht vom Display erfasst werden, schwarz.

Das HMD wird im Raum getrackt, was in Software für eine Translation und Rotation der Kamera innerhalb einer Szene genutzt werden kann, um die Kopfbewegung des Nutzers zu erfassen. Das Erfassen der Extrinsik geschieht dabei ohne spürbare Latenz und mit hoher Frequenz von 225Hz über Trackingsensoren, welche am HMD angebracht sind. Dadurch kann die Interaktivität und korrekte Positionierung der virtuellen Kamera auch während der Bewegung des Benutzers stets gewährleistet werden.

3.2 Voraussetzungen durch Software

Für die Implementierung unserer VR-Software haben wir uns für C++ in Kombination mit den Frameworks VTK ¹ und Vive-SDK entschieden. Die Wahl der Programmiersprache fiel auf C++, da wir damit bereits Erfahrung haben und es entsprechende Schnittstellen in beiden genannten Frameworks gibt. VTK ist eine Softwarebibliothek, die viele nützliche Funktionen für das Erstellen von 3D-Visualisierungen besitzt. In der Version 7.1.1 (erschienen am 20.03.2017) von VTK wurde der VR-Support eingebaut, sodass Position und Orientierung von Controller und HMD mittels Funktionsaufrufen von VTK ausgewertet werden können. Darüber hinaus benutzen wir VTK für das Laden von Datensätzen und das Rendering. Die in dem Toolkit implementierten Renderer haben einen einfachen Level-Of-Detail Ansatz, auf den wir uns im Rahmen dieses Projektes verlassen, um ein ausreichend schnelles Rendering zu ermöglichen.

3.3 Anforderungen

Die Aufgabenstellung erfordert einen Vergleich der Perzeption sowie Qualität und Effizienz von Interaktionen zwischen 2D-Monitor und aktuell verfügbaren HMDs. Dadurch ergeben sich für eine VR-Software folgende Anforderungen:

Visualisierung der Eingabedaten

In unserer Software soll es möglich sein, medizinische Bilddaten darzustellen. Diese Bilddaten, welche typischerweise als Schichtstapel vorliegen, müssen so visualisiert werden, dass unterschiedliche Strukturen wahrnehmbar sind.

¹The Visualization Toolkit, <http://www.vtk.org>, zuletzt abgerufen: 29.05.2017

Virtuelle Umgebung

Um die Bilddaten nicht in einen unendlich weit wirkenden Raum zu platzieren, soll darauf geachtet werden, dass eine virtuelle Umgebung erstellt wird, die den Nutzer nicht ablenkt und Ängsten wie Akrophobie (Angst vor Höhe) und Klaustrophobie (Angst vor Einengung) entgegenwirkt.

Basisinteraktionen

Um einen geladenen Datensatz explorieren zu können, müssen Basisinteraktionen wie Rotation, Translation und Skalierung auf dem Datensatz ermöglicht werden. Darüber hinaus soll die Kamera beweglich sein und der Bewegung des Kopfes folgen.

Tools

Um einen Vergleich zu ermöglichen, welcher über Basisinteraktionen hinausgeht, werden Tools benötigt, die einfache Explorations- und Vermessungsaufgaben ermöglichen und in ähnlicher Form in der Vergleichssoftware für 2D-Monitore existieren.

3.4 Ideen und Konzepte

Basierend auf den zuvor beschriebenen Anforderungen werden nun Ideen und Konzepte vorgestellt. Diese dienen als Grundlage für die Implementierung, welche im nächsten Kapitel beschrieben wird.

Visualisierung der Eingabedaten

Für die Visualisierung der Eingabedaten sehen wir zwei Möglichkeiten, die man anhand der Art der vorliegenden Daten unterteilen kann: Zum Einen lassen sich die rohen Volumendaten verarbeiten. Diese entsprechen der direkten Ausgabe eines bildgebenden Verfahrens (z.B. Computertomographie). Die Darstellung von Volumendaten setzt voraus, dass der Renderingprozess mit Hilfe eines Volumenrenderers durchgeführt wird. Dieses Renderingverfahren ist aufwändig, aber mit heutigen Grafikkarten auch mit hoch aufgelösten Datensätzen mit flüssiger Bildrate möglich. Zum Anderen besteht die Möglichkeit, segmentierte Daten, welche in Form von Oberflächenmodellen vorliegen, direkt mithilfe einer *gängigen* Renderingpipeline (Rasterisierung) darzustellen.

Virtuelle Umgebung

Mit Start der Software befindet sich der Nutzer in einem virtuellen Raum. Damit ein Bezug zur Realität hergestellt wird, soll es sich nicht um einen leeren 3D-Raum handeln, sondern um einen Raum mit Wänden und einem Fußboden - der *Showroom*. In diesem Raum soll der Datensatz vor dem Nutzer schweben. Durch die Kalibrierung des Systems bezüglich der realen Umgebung im Aktionsraum – das umfasst Positionen von Hindernissen, wie Tische, aber auch die Position des Bodens –, können wir dem Nutzer den Datensatz zu Beginn in Originalgröße präsentieren. Dies soll bereits einen ersten guten Eindruck von den Größenverhältnissen innerhalb eines Datensatzes ermöglichen, wie es mit einem herkömmlichen Monitor-Setup nur möglich ist, wenn der Monitor größer ist als die darzustellenden Daten. Sobald die Szene komplett geladen ist, soll man sich in der Szene bewegen können, indem man auf natürliche Weise geht. Dies ermöglicht das Betrachten des Datensatzes von allen Seiten.

Basisinteraktionen

Die Basisinteraktionen – Rotation, Translation und Skalierung – sollen es dem Nutzer ermöglichen den Datensatz so zu transformieren, dass für die nötigen Aufgaben korrekt ausgerichtet ist. Ziel ist eine 3D-Interaktion, die einfach ausgeführt werden kann und dabei schnell zum gewünschten Erfolg führt. Der Benutzer soll dafür den Datensatz mithilfe der Controller an beliebigen Punkten anfassen können. Die dabei angefassten Punkte sollen bei der Interaktion wiederum immer mit dem Controller mitgeführt werden. Wurde ein Punkt einmal angefasst, bewegt sich dieser mit dem entsprechenden Controller mit, bis er losgelassen wird. Die Ausrichtung des Datensatzes wird ebenso dem Controller angepasst. Falls der Nutzer den Datensatz mit beiden Controllern an verschiedenen Punkten anfassen, wird der Datensatz mithilfe von Rotation, Skalierung und Verschiebung so angepasst, dass beide Punkte stets den Controllern folgen.

Tools

Unser Prototyp soll drei verschiedene Tools beinhalten, die es ermöglichen den Datensatz besser zu explorieren und nötige Informationen aus ihm zu extrahieren. Diese Tools sollen neben den Basisinteraktionen immer auswählbar sein. Ein Feature soll dabei sein, dass sie an einen beliebigen Controller gebunden werden können.

So gibt es verschiedene Varianten, die Tools auszuwählen. Eine Variante wäre es, die Tools als greifbare Gegenstände in die Szene zu integrieren und den Toolwechsel durch das Greifen eines anderen Tools durchzuführen. Hier kommen allerdings Probleme auf, zum Beispiel, dass man sich immer erst an einen bestimmten Ort bewegen muss, um die Tools auszuwählen oder dass Tools irgendwann überall im Raum schweben, wenn diese nicht automatisch entfernt werden.

Wir haben uns für ein Toolauswahl mittels Menü entschieden, das die Auswahl eines Tools für einen Controller ermöglicht. Das Menü soll zudem immer gut zu erkennen sein. Da die VR-Brille eine begrenzte Auflösung hat, muss das Menü eine gewisse minimale Größe besitzen, um auf dem HMD-Display ausreichend aufgelöst zu sein. Entsprechend bietet es sich an, das Menü in der Nähe des Nutzers zu platzieren. Deswegen soll das Menü an die Controller gebunden sein, da die Controller bei Benutzung der Software immer in der Nähe des HMDs sind. Weiterhin ist dadurch die Position des Menüs vom Benutzer kontrollierbar und an keine feste Position im Raum gebunden, die vom Nutzer regelmäßig aufgesucht werden müsste. Da die Auflösung nicht genügend Spielraum bietet um Schriften genügend aufzulösen, sollen die Menüpunkte durch Icons visualisiert werden. Innerhalb des Menüs kann das gewünschte Tool ausgewählt werden. Das somit selektierte Tool wird anschließend an den Controller gebunden. Die Tools sollen immer durch die gleiche Taste ausgelöst werden. Die Auswahl der Tools soll wiederum immer über das Menü geschehen. Im Folgenden werden die Tools und ihre Eigenschaften genauer vorgestellt.

Das Home-Tool bringt den Datensatz zurück in seine ursprüngliche Position, Ausrichtung und Skalierung. Dadurch kann der User zu einem stabilen und bekannten Zustand der Szene zurückkehren, falls er die Orientierung verloren hat. Sollte es passieren, dass das Tracking der Controller bei der Interaktion gestört worden sein, so kann der Datensatz immer wieder in einen bekannten Zustand überführt werden.

Das Clipping-Plane-Tool soll es ermöglichen den Datensatz entlang einer frei definierbaren Ebene zu schneiden. Dabei soll der Nutzer die Möglichkeit haben eine Ebene hinzuzufügen und wieder zu entfernen. Zudem sollte die Ebene am Datensatz fest arritierbar sein, sodass man den Datensatz anschneiden kann und danach frei weiterbearbeiten kann, ohne die Ebene nachführen zu müssen.

Das Measurement-Tool dient dazu, Distanzen im Datensatz zu vermessen. Hierbei sollen lineare Abstände gemessen werden. Dazu werden im Datensatz 2 Punkte markiert, deren Abstand zueinander vermessen werden soll. Anschließend soll die Länge angezeigt werden. Falls ein oder mehrere falsche Punkte gesetzt worden, soll es möglich sein diese zu editieren oder die gesamte Messung zu löschen. Zudem soll es möglich sein mehrere Messungen innerhalb des Datensatzes zu platzieren um später zum Beispiel verschiedene Messungen gegeneinander auszuwerten. Falls zum Beispiel nicht direkt optisch entschieden werden kann, ob eine Strecke geringer als die andere ist, so kann mithilfe von mehreren Messungen dies besser verglichen und entschieden werden. Damit die Vermessungspunkte präzise platziert werden können, muss die virtuelle Präsenz des Controllers so angepasst werden, dass solch ein Punkt präzise gesetzt werden kann.

4. Implementierung

Visualisierung der Eingabedaten

Am Anfang des Projektes haben wir mit dem Volumenrendering experimentiert. Da sich die von uns eingesetzte Renderingbibliothek für Volumenrendering in Kombination mit VR noch im experimentellen Status befand, war das Ergebnis des Renderings stark verzerrt. Durch die immersive, stereoskopische Darstellung wirkt sich ein fehlerhaftes Rendering viel stärker aus als bei 2D-Monitoren und kann Unwohlsein beim Benutzer hervorrufen. Wir vermuten hier, dass der Grund für den Fehler darin besteht, dass VTK bestimmte Kameraparameter wie Position und Winkel der zwei Kameras für die beiden Augen zueinander falsch setzt. Eine tiefere Analyse der Shader ist hier notwendig, um den Fehler im VR-Modus des Volumenrenderings innerhalb der von uns eingesetzten VTK-Version zu finden. Somit ist das vorhandene Volumenrendering für dieses Projekt nicht nutzbar, da es uns nicht möglich war, eine schnelle Korrektur des Problems bereitzustellen. Wir haben uns deshalb dafür entschieden, das Rendering auf Basis von Meshes durchzuführen. Dabei haben wir anschließend darauf geachtet, dass alle Funktionalitäten auch auf ein Rendering mit Volumendaten übertragbar sind. Sobald das Rendering für Volumendaten mittels Darstellung in VR-Systeme in VTK korrekt implementiert ist, lässt sich die Renderingpipeline unserer Software ohne großem Aufwand austauschen. In diesem Fall stehen die gleichen Funktionalitäten bereit, wie bei der Darstellung von Oberflächenmodellen. Die einzige notwendige Anpassung besteht in einer Implementierung für das Einstellen von Window/Level. Das Setzen von Window/Level dient u. A. dazu, die Darstellung auf gewisse Wertebereiche zu beschränken und somit für eine spezifische Situation ungewünschte Materialien, wie Luft, Weichgewebe, Knochen, etc., ein- oder auszublenden. Damit können bestimmte Strukturen hervorgehoben werden oder ausgewählten Wertebereichen ein höherer Grauwertkontrast gegeben werden.

Das einfache Abändern der Renderingpipeline ist möglich, da jedes gerenderte Objekt in VTK ein sogenanntes `vtkProp3D` ist. Diese Klasse beinhaltet alle 3D-Positionierungsfunktionen. Da wir direkt auf dieser Klasse aufbauen, können wir hier das Oberflächenmodell einfach durch ein Volumen ersetzen. Alle Interaktionen bleiben anschließend erhalten und das Volumen kann genauso bearbeitet werden wie das Mesh. Tests hierzu haben stattgefunden und das Verschieben und Clippen eines Volumendatensatzes hat bereits funktioniert, mit den bereits beschriebenen Fehlern beim Rendering.

Showroom für medizinische Datensätze

Der Showroom wurde in einer virtuellen Größe von 10m x 10m angelegt. Der Nutzer hat dadurch einen gewissen Eindruck von Weite, aber trotzdem nicht das Gefühl in einem unendlichen weiten, leeren Raum zu stehen. Zudem wird im späteren Verlauf die Skalierungsinteraktion vorgestellt. Dadurch kann der Datensatz auf ein Vielfaches seiner Größe anwachsen. Ein kleinerer Raum hätte zur Folge, dass der Datensatz in skaliertem Form sehr zeitig abgeschnitten werden könnte. Der Boden wurde mit einer

Kachelung von schwarzweißen Fliesen texturiert. Durch die vielen dadurch erzeugten Linien fällt die Orientierung leichter, zudem wird durch die schwarzweiße Farbgebung ein Boden geschaffen, der gute Kontraste zum betrachteten Datensatz bildet. Der Bezugspunkt zu den Wänden hilft zudem dabei, sich im Raum zu orientieren. Die Wände wurden wiederum mit einem Fliesenmuster versehen, sodass auch diese optische Bezugspunkte bieten, um eigene Bewegungen innerhalb der Szene besser verarbeiten zu können. In der Mitte des Raums wird der Datensatz platziert. Nachdem die Szene und der Datensatz vom Programm geladen wurde, wird die Szene auf der VR-Brille angezeigt. Die VR-Brille wird standardmäßig so kalibriert, dass der Nullpunkt und somit auch der Anfangspunkt des Datensatzes in der Mitte der für die VR-Brille kalibrierten Bereiches liegt. Somit ist es nach dem Laden direkt möglich, sich um den Datensatz zu bewegen und einen ersten Eindruck des Datensatzes zu bekommen. Da die originale Größe des Datensatzes als Standardwert genutzt wird, entspricht der erste Eindruck, den der Nutzer vom Datensatz bekommt, der realen Größe der gegebenen Struktur. Der Datensatz liegt zudem in einer eigenen Transformationsgruppe. Wird diese Gruppe transformiert, so werden die Transformationen nur auf Objekte dieser Gruppe angewandt und nicht auf andere Objekte, z.B. den Raum. Spätere Annotationen wie Messpfeile oder die Clippingebene werden ebenso zu dieser Transformationsgruppe hinzugefügt, sodass diese sich innerhalb des Datensatzes immer am gleichen Ort befinden. Die Transformationsgruppe beinhaltet eine einfache 4x4-Matrix, die die Transformation der Objekte darin festlegt. Hier kann also jegliche Kombination von Rotation, Translation und Skalierung mithilfe homogener Koordinaten ausgedrückt werden.

Basisinteraktionen

Sobald man sich in der Szene befindet, lassen sich mit den Controllern Basisinteraktion, d.h. Rotieren, Skalieren und Translieren, auf dem Datensatz ausführen. Die Aktionen werden dabei durch Betätigen der Triggerbuttons ausgelöst. Drückt man auf einem Controller den Triggerbutton, so startet man die Translation und Rotation des Datensatzes, wobei beide Aktionen der Bewegung des genutzten Controllers folgen. Dabei ist es unabhängig, ob der Controller den Datensatz berührt oder außerhalb davon ist. Das Rotationszentrum befindet sich dabei immer im Controller. So kann der Datensatz von beliebigen Stellen auch angefasst und frei bewegt werden. Die erzeugte Transformation wird auf die gesamte Transformationsgruppe des Datensatzes angewendet. Dabei gibt der Controller die genaue Transformation vor, d.h. der Controller *bindet* sich an den Datensatz und lässt sich innerhalb des Datensatzes nicht mehr verschieben. Wird der Controller gedreht oder bewegt, so wird diese Transformation direkt auf die gesamte Transformationsgruppe angewendet. Darüber hinaus kann man die Skalierung des Datensatzes initiieren, indem man an dem zweiten Controller den Triggerbutton drückt und die Controller zueinander oder voneinander weg bewegt. Sobald man einen Triggerbutton während einer Skalierung loslässt, wird die zu diesem Zeitpunkt vorhandene Skalierung für die Transformationsgruppe beibehalten. Während der Skalierung kann der Datensatz zudem rotiert und verschoben werden. Dabei werden die beiden Punkte, die mit den Controllern angefasst werden, als feste Referenzpunkte am Datensatz genutzt. Die Rotation kann ebenso durchgeführt werden, wenn beide Controller den Datensatz anfassen. Dabei wird die Rotation folgendermaßen berechnet: Zuerst wird die

Basislinie zwischen den beiden Controllern bestimmt. Dies ist die erste Achse der Rotation. Zudem besitzen beide Controller eine eigene Ausrichtung. Hierbei wird von beiden Controllern der Vektor genommen, in denen die Controller zeigen. Diese beiden Vektoren werden gemittelt. Daraus ergibt sich der zweite Vektor für die Rotationsabbildung. Der letzte Vektor ergibt sich aus dem Kreuzprodukt dieser beiden Vektoren, da die Rotationsabbildung orthogonal ist. Diese drei Vektoren werden orthogonalisiert und normalisiert und ergeben zusammen die Rotationsabbildung, die das derzeitige System beschreibt. Bestimmt man nun den Rotationsunterschied zwischen Zustand bei der Initialisierung der Interaktion und aktuellem Zustand erhält man die Rotation, die während der Skalierungsinteraktion ausgeführt wurde. Der Datensatz kann damit auf eine natürliche Art transformiert werden.

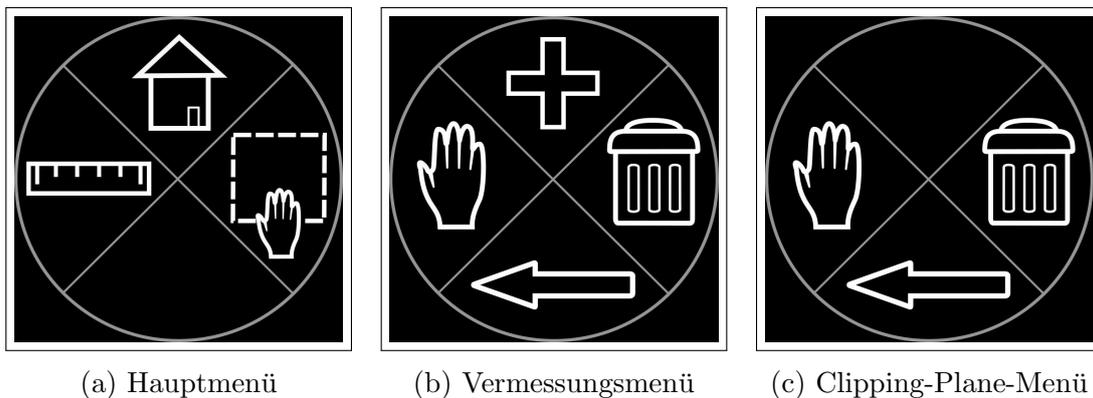
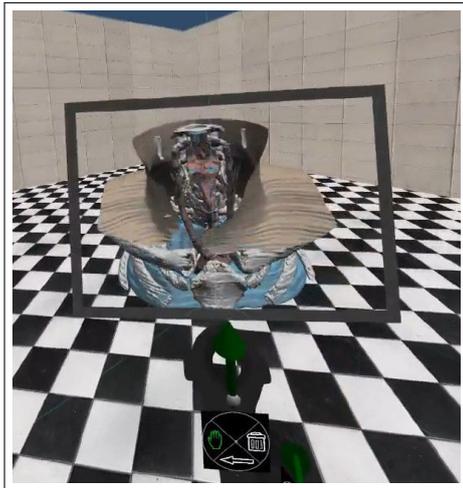


Abbildung 4.1: Menüs, die über das Touchpad der Controller gesteuert werden

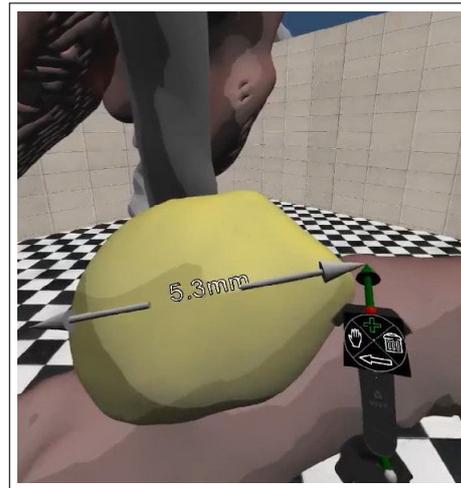
Tools

Auf dem Touchpad wird zu jeder Zeit ein Menü dargestellt, siehe [Abbildung 4.1](#). Dieses Menü gibt Zugang zu den unterschiedlichen Tools, die zur Exploration und Vermessung eingesetzt werden können. Die Symbole sind dabei in einem einfachen Linienstil gehalten mit identischen Linienstärken für Außen- und Innenlinien. Eine detailarme Darstellung kombiniert mit einem hohen Kontrast soll es ermöglichen, die Grafiken auch noch mit wenigen Pixeln zu erkennen, beispielsweise, wenn die Controller in einem flachen Winkel zur Betrachtungsposition gehalten werden. Die Implementierung der im Konzept vorgestellten Tools wird nun im Folgenden erläutert:

Das Measurement-Tool dient zur Vermessung von Strukturen im Datensatz. Mithilfe dieses Werkzeugs werden die Messungen in Form von Messpfeilen durchgeführt. Das Tool hat wie im Konzept verschiedene Modi. Im ersten, automatisch ausgewählten Modi ist es möglich, Pfeile in der Szene hinzuzufügen. Die Interaktion entspricht dabei der einer Gummiband-Metapher (engl. *rubber band metaphor*), sodass ein intuitives *Aufziehen* der Messpfeile ermöglicht wird. Am Pfeil selbst ist zu jeder Zeit der Interaktion die vermessene Länge über einen Text zu erkennen. Im Datensatz können so mehrere Messungspfeile platziert werden. Die angezeigte Länge entspricht dabei stets der Länge in Datensatzkoordinaten - d.h. die Länge ist unabhängig von einer Skalierung der Szene. Der zweite Modus des Vermessungstools ist



(a) Clipping



(b) Vermessung

Abbildung 4.2: Werkzeuge

es Pfeile zu justieren. Falls auffällt, dass einer oder beide Punkte eines Vermessungspfeils nicht korrekt gesetzt sind, so ist es in diesem Modus möglich die Punkte durch erneutes Anfassen neu zu positionieren. Der dritte Modi stellt die Möglichkeit dar, Pfeile aus der Szene zu entfernen. Dabei kann der Nutzer die jeweiligen zu löschenden Pfeile auswählen und dann mit Drücken des Triggerbuttons entfernen. Damit alle Selektions- und Positionierungsinteraktionen präzise ausgeführt werden können, wurde die Darstellung des virtuellen Controller durch Hinzufügen einer Pfeilspitze angepasst. Diese macht es dem Nutzer möglich, exakte Positionen für Pfeile zu setzen.

Das Home-Tool kann im Hauptmenü durch Druck auf ein symbolisches Haus (Abbildung 4.1a, oben) ausgeführt werden. Dabei wird die gesamte Transformationsgruppe auf die Einheitstransformation zurückgesetzt und der Datensatz erscheint wieder dort, wo er zu Beginn des Programms geladen wurde.

Das Clipping-Plane-Tool erlaubt das Schneiden des Datensatz mit einer frei positionierbaren Ebene (engl. *clipping plane*). Somit wird es möglich in den Datensatz zu blicken und Strukturen, die die Sicht behindern, zu entfernen, welche die Sicht versperren. Da die Ebene interaktiv positioniert wird, lässt sich direkt das Resultat des Schnittes der Clipping-Plane mit der Szene sehen, siehe Abbildung 4.2a. Die Clipping-Plane lässt sich festsetzen, sodass ein Schnitt an der gleichen Stelle im Datensatzkoordinatensystem unabhängig von der Rotation, Translation oder Skalierung des Datensatzes bleibt. Zudem kann die Position der Ebene geändert werden, oder die Ebene komplett aus der Szene entfernt werden. Derzeit steht maximal eine Clipping-Plane für eine Szene zur Verfügung.

5. Evaluierung

5.1 Ziel

Die Evaluierung soll mögliche Unterschiede zwischen den bisher genutzten 2D-Interfaces und dem von uns implementierten VR-Interface aufzeigen. Dafür wurde eine Nutzerstudie durchgeführt. Diese Studie soll zum einen die Geschwindigkeit und Genauigkeit der Interaktionen ermitteln, und zum anderen die subjektive Wahrnehmung seitens der Probanden bezüglich der Interaktionen darlegen.

5.2 Software-Vergleichskandidat

Die Software, die wir als Vergleichskandidaten nutzen, ist der Dornheim Segmenter Analyzer 2016.10. Dabei handelt es sich um eine Visualisierungs-, Analyse- und Segmentierungssoftware für medizinische Daten. Der Dornheim Segmenter Analyzer erlaubt die Arbeit mit Volumendaten und Oberflächenmodellen. Zu den Werkzeugen gehören viele 3D-Bearbeitungs- und Segmentierungswerkzeuge, aber auch Analyse-tools, u.A. das Vermessen von Längen in 2D und 3D.

5.3 Aufgaben

Für die Nutzerstudie haben wir drei Aufgaben gewählt, wobei eine Aufgabe ausschließlich der Auswertung der Genauigkeit der Basisinteraktionen in der VR-Umgebung dient. Die beiden anderen Aufgaben behandeln das Vermessen von gegebenen Strukturen. Die zwei Vermesseraufgaben werden sowohl im Dornheim Segmenter Analyzer, als auch in unserem VR-Prototyp durchgeführt, jeweils mit dem gleichen Datensatz durchgeführt. Mit jedem Teilnehmer der Studie haben wir die Reihenfolge der Systeme abgewechselt. Die eine Hälfte kam zuerst mit dem Dornheim Segmenter Analyzer in Berührung, die Anderen absolvierten die Aufgaben zuerst mit dem VR-Prototyp.

Einarbeitung in die VR-Umgebung

Um die Nutzer an die ungewohnte VR-Umgebung und -Interaktion zu gewöhnen haben wir vor den Tests am VR-Prototyp jeden Nutzer einen Demoraum präsentiert. Dieser Raum beinhaltet ein 3D-Modell, mit dem die Nutzer alle Tools und Möglichkeiten der VR-Software ausprobieren konnten. Sobald die Nutzer sich sicher in der VR-Umgebung gefühlt haben, wurde mit den Aufgaben begonnen. Falls den Testprobanden die 2D-Umgebung noch nicht bekannt war, gab es auch hier eine Einführung, bei der die Testprobanden alles ausprobieren konnten.

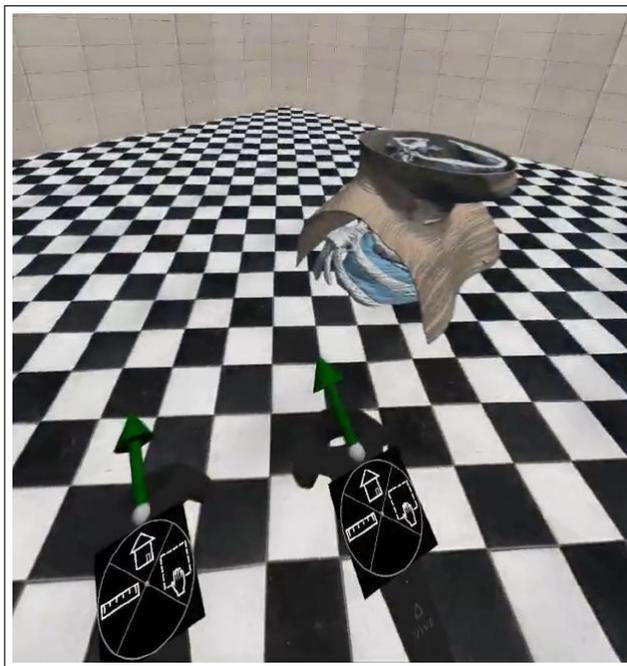


Abbildung 5.1: Geladener Datensatz in der VR-Umgebung

VR-Interaktionsaufgabe

Diese Aufgabe wurde nur in der VR-Umgebung durchgeführt. Ziel ist es hierbei zu überprüfen, ob die Nutzer das Gefühl haben alle Interaktionen immer wie gewünscht durchführen zu können und wie schnell die Steuerung von den Nutzern angenommen wird. Die zu lösende Aufgabe ist es, eine transformierte Kopie eines Modells deckungsgleich über die Originalversion des Modells zu legen. Dabei liegt das zu bewegendes Modell verändert in Skalierung, Rotation und Position vor. Durch einfache Griffoperationen mit den Controllern soll das Modell mit dem anderen Modell überlagert werden. Sobald der Proband mit dem Ergebnis zufrieden ist, wird der Test beendet. Anschließend gibt das Programm die Anzahl der getätigten Interaktionen, sowie Skalierungs-, Rotations- und Translationsfehler zum Vergleichsmodell aus.

Vermessungen innerhalb medizinischer Daten

Für die Vermessungsaufgaben wird ein segmentierter medizinischer Datensatz genutzt. Dabei handelt es sich um eine CT-Aufnahme, die den Bereich des Halses abdeckt (siehe [Abbildung 5.1](#)). Markant für diesen Datensatz ist eine Stenose der Trachea. Folgende Strukturen liegen in dem Datensatz segmentiert vor: Einzelne Lymphknoten, die Haut, Knochen, Hauptschlagadern und der innere und äußere Abschluss der Trachea.

Das Vermessen von Lymphknoten beinhaltet das grundlegende Explorieren des Datensatzes, sowie den Lymphknoten zu finden, der am nächsten an einer der Hauptschlagadern liegt. Ist dieser gefunden, soll der kürzeste Abstand zur Hauptschlagader gemessen werden und anschließend die Hauptausdehnungsachsen des Lymphknotens selbst. Für das Absolvieren dieser Aufgabe muss ein grobes mentales

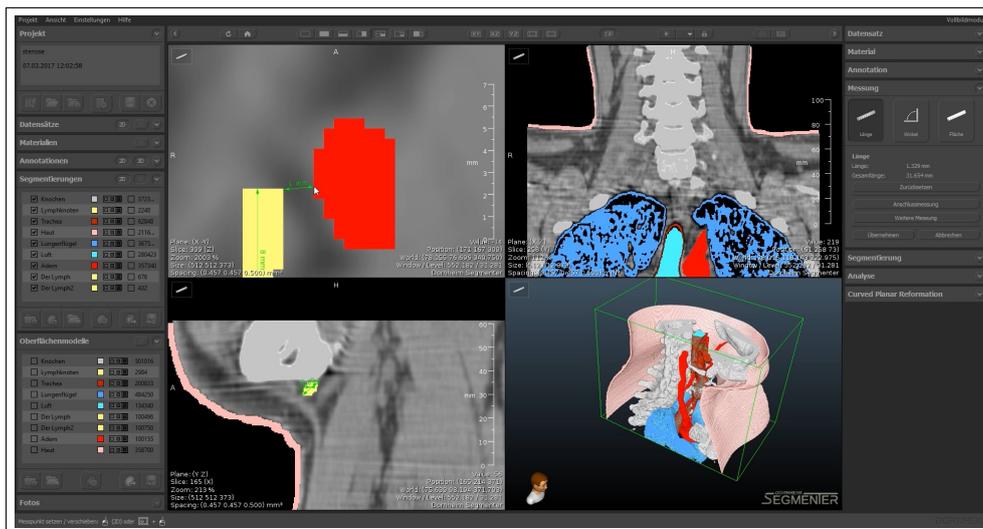


Abbildung 5.2: Manuelles Vermessen eines Lymphknotens im Dornheim Segments

Modell des Datensatzes entstehen, mit Fokus auf die Lage der Hauptschlagadern, sowie aller Lymphknoten in Bezug auf die Hauptschlagadern. Die drei zu betrachtenden und zu vergleichenden vorsegmentierten Lymphknoten liegen dabei alle an einer Hauptschlagader. Sobald alle drei Lymphknoten lokalisiert wurden, müssen die Abstände der einzelnen Lymphknoten zu der Hauptschlagader verglichen werden. Auswertungsziel dieser Aufgabe ist es, die Geschwindigkeit beim Explorieren in drei Dimensionen inklusive einfacher Vermessungsaufgaben für die eingesetzten Technologien zu ermitteln.

Das Vermessen der Stenose erfordert vom Nutzer die engste Stelle der Trachea zu finden und zu vermessen. Um diese Aufgabe zu bewältigen, kann auf die mentale Karte aus der vorherigen Aufgabe zurückgegriffen werden. Ähnlich zur vorherigen Aufgabe muss eine Stelle als dünnste Stelle innerhalb der Trachea erkannt werden. Dabei kann auf das Augenmaß gesetzt werden, oder bei ähnlich eng erkannten Stellen eine Messung durchgeführt werden, um die engste Stelle zu ermitteln. Da die Stenose im Datensatz sehr markant ist, wird erwartet, dass die Teilnehmer die entsprechende Stelle per Augenmaß finden und im Zielbereich Messungen durchführen, um die wirklich engste Stelle zu ermitteln.

5.4 Auswertung

An der Nutzerstudie nahmen sieben Personen teil. Alle ermittelten Messwerte sind im Anhang tabellarisch aufgelistet.

VR-Interaktionsaufgabe

Bei der Interaktionsaufgabe haben die Probanden unterschiedliche Herangehensweisen aufgezeigt. Während einige durch mehrmaliges Anfassen des Modells die Aufgabe gelöst haben, nutzten andere Personen das direkte visuelle Feedback der VR-Brille und haben das Modell direkt an den Punkt gezogen und skaliert, sodass sie nur sehr

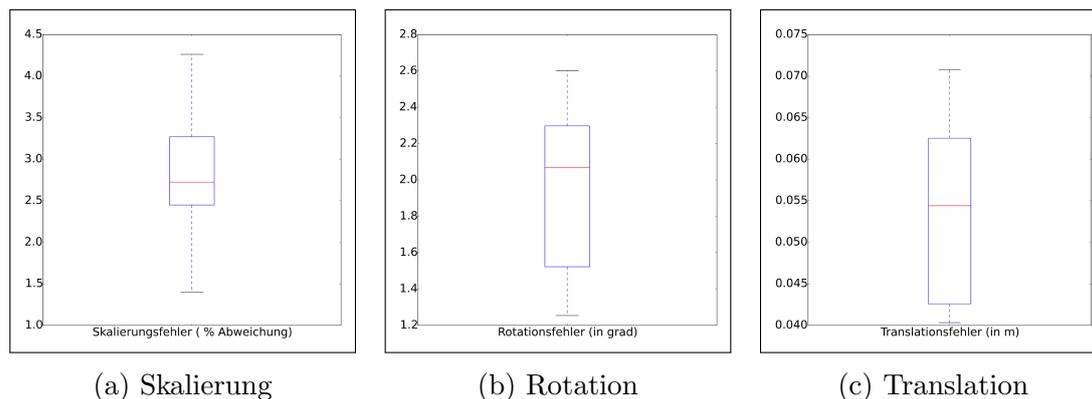


Abbildung 5.3: Abweichungen der VR-Interaktionsaufgabe

wenige Interaktionen brauchten. Somit zeigten sich hier zwei verschiedene Interaktionsmuster: Muster A (3 Probanden), bei dem Schritt für Schritt die Aufgabe gelöst wird und Muster B (4 Probanden), welches das direkte Verschieben des Modells an die Endposition darstellt. (siehe Interaktionsdaten im Anhang)

Nachdem der Proband die Modelle übereinander gelegt hat, wertet das System den Skalierungs-, Rotations- und Translationsfehler aus, siehe [Abbildung 5.3](#). Die Skalierung wurde dabei im Schnitt um 2,7% von der optimalen, d.h. identischen Größe, verfehlt. Bei der Rotation wurde ein durchschnittlicher Fehler von 2,1° gemacht. Der Translationsfehler liegt im Durchschnitt bei 5,5mm.

Der Fehler bezüglich Skalierung und Rotation entspricht unseren Erwartungen und erscheint uns als ausreichend präzise für übliche Aufgaben in einer VR-Umgebung. Der Translationsfehler dagegen ist größer, als wir erwartet haben, mit über 5mm Abweichung im Schnitt. Unser Test hat allerdings von den Probanden zusätzlich zu den notwendigen Interaktionen das Abschätzen der Lage der beiden Modelle zueinander abverlangt. Da beide Modelle opak gerendert wurden und keinerlei Rückmeldung bezüglich der Abweichungen während des Tests gegeben wurde, war bereits die Entscheidung bezüglich der Ähnlichkeit beider Modelle schwierig. Ein zukünftiger Test könnte hier verbessert werden, indem beispielsweise ein Oberflächenmodell halbtransparent gerendert würde, sodass man eine Überlappung beider Modelle besser erkennen kann.

Vermessung von Lymphknoten

[Abbildung 5.4](#) zeigt, welche Interaktionen für das Vermessen der Lymphknoten genutzt wurden. Hier ist zu sehen, dass einige Probanden auf verschiedene Arten gearbeitet haben. Es gab beispielsweise einen Probanden, der die Clippingebene nicht verwendet hat und stattdessen direkt in das Volumen eingetaucht ist. Weiterhin lässt sich erkennen, dass gesetzte Messpfeile oft nachjustiert werden, anstatt diese komplett erneut zu setzen. Der Homebutton wurde insgesamt für diese Aufgaben lediglich einmal genutzt. Die Kategorien "Messpfeil versucht zu bewegen" und "Messpfeil versucht zu entfernen" zeigen die Anzahl der missglückten Versuche die jeweilige Interaktion mit einem Pfeil durchzuführen. Hier erkennt man, dass relativ wenige missglückte Versuche nötig waren, bis die gewünschte Operation ausgeführt wurde. Dies spricht zumindest dafür, dass sich die Selektion der im VR-System so verhält, wie die Probanden es erwartet haben.

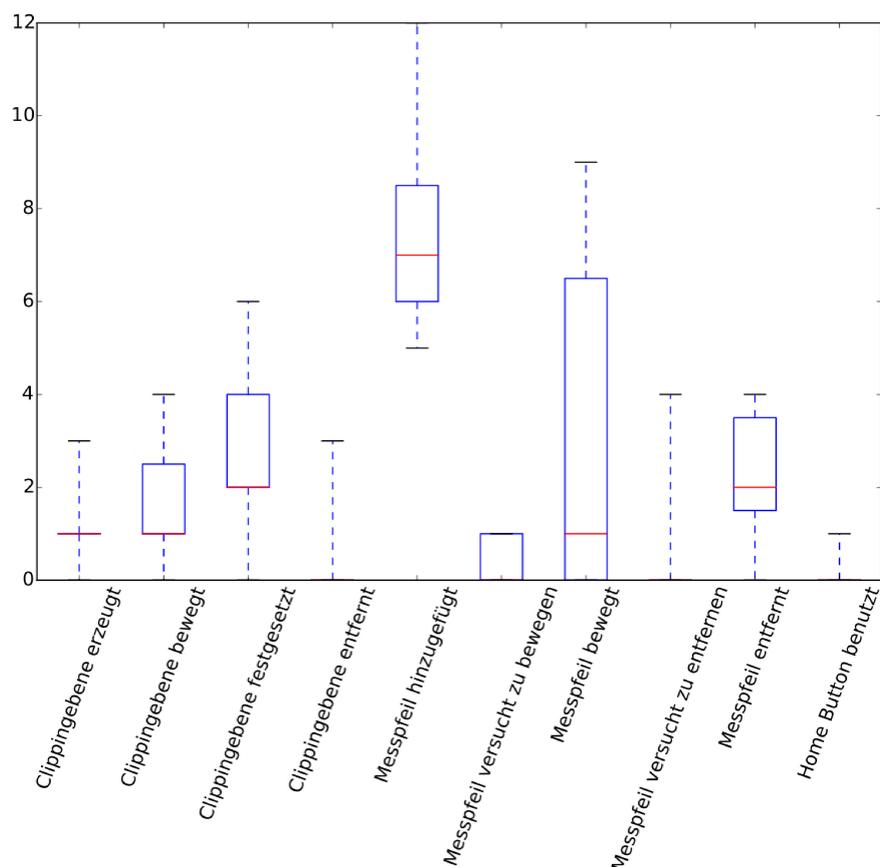


Abbildung 5.4: Anzahl der Toolinteraktionen, die für die Lymphknotenvermessung gebraucht wurden

Vermessung der Stenose

Abbildung 5.5 zeigt, dass bei dieser Aufgabe jeder Proband mit der Clippingebene gearbeitet hat – bei dem zuvor beschriebenen Lymphknotentest hat ein Proband auf die Clippingebene verzichten können. Da das Ziel darin bestand, das Innere der Trachea zu vermessen, war bei dieser Aufgabe jedoch ein Schnitt mit der Clippingebene durch die Szene notwendig. Erwähnenswert ist, dass die Messpfeile bei den meisten Probanden selten neu gesetzt wurden, aber auch nicht justiert wurden. Die einmalig ausgewählten Stellen schienen also bereits der Vorstellung der Probanden zu entsprechen. Deswegen kam es hier zu wenig bis keine fehlgeschlagenen Interaktionsversuche mit den Messpfeilen.

Messergebnisse und Messunterschiede

Um einen Vergleich der Genauigkeit durchzuführen, haben wir bei jedem Test die Messergebnisse der Probanden abgespeichert. Da uns keine *Ground Truth* zur Verfügung steht, führen wir in Abbildung 5.6 die Differenzen zwischen dem 2D-System und dem VR-System auf. Die Unterschiede der Messergebnisse zwischen 2D- und VR-System betragen maximal 0.6mm. Zu erkennen ist, dass die Unterschiede größer werden, wenn die zu vermessenden Strukturen kleiner werden. Die größte Ausdehnung des Lymphknotens wurde zum Beispiel mit einer vergleichsweise kleinen Abweichung

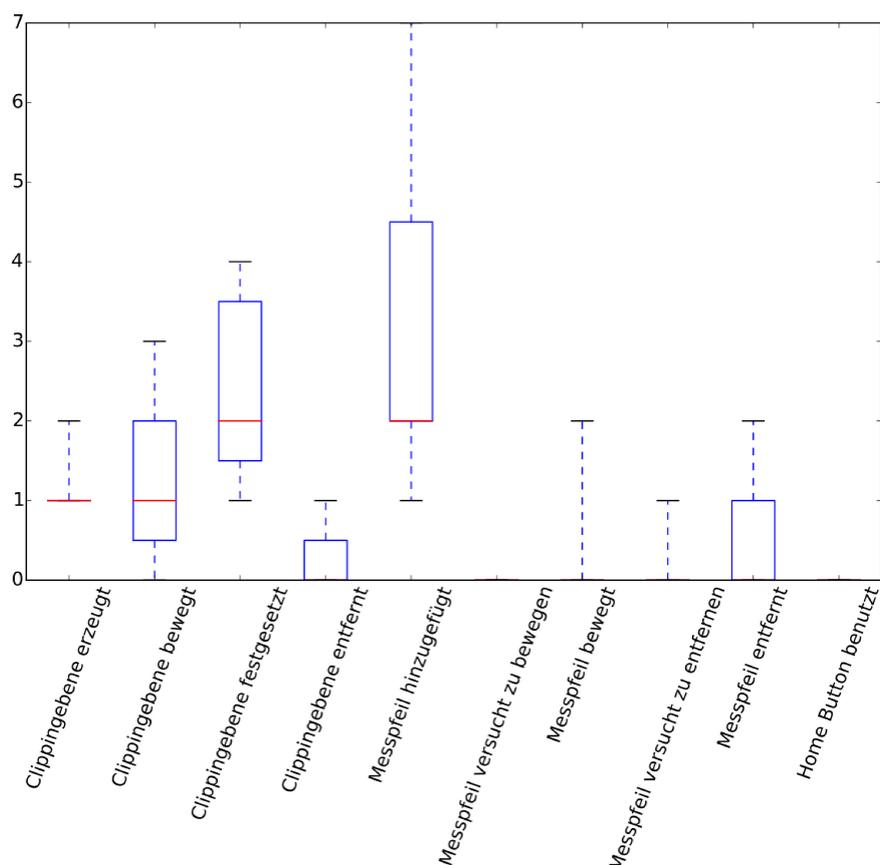


Abbildung 5.5: Anzahl der Toolinteraktionen, die für die Stenosevermessung gebraucht wurden

im Mittel gefunden. Die minimale Ausdehnung des Lymphknotens hingegen führt zu großen Unterschieden zwischen 2D-System und VR-System. Dies kann daran liegen, dass die Nutzer im VR-System einen ganz anderen 3D-Eindruck haben und die Größenverhältnisse entsprechend anders einschätzen. Weiterhin ist hier anzumerken, dass die Auflösung des Datensatzes 1mm umfasst. Das Rendering des 2D-Systems zeigt hierbei stärker die Grenzen der Auflösung des Volumendatensatzes, da hierbei neben segmentierten Oberflächenmodellen auch der unbearbeitete Datensatz mittels Volumenrendering dargestellt wird. In VR standen lediglich segmentierte Oberflächenmodelle zur Verfügung, die die Oberfläche automatisch interpolieren und auch eine höhere Genauigkeit vortäuschen.

Zeiten für die Aufgaben

In [Abbildung 5.7](#) ist zu erkennen, dass die Aufgaben in VR im Durchschnitt schneller gelöst wurden als in der 2D-Software. Das Auffinden und Vermessen der Lymphknoten hat in der VR-Umgebung nur die Hälfte der Zeit benötigt. Auch die Messaufgaben bezüglich der Stenose wurden i.d.R. von den Probanden in der VR-Umgebung schneller gelöst, mit einer im Schnitt 25% schnelleren Bearbeitungszeit. Insgesamt zeigt sich ein deutliches Bild zugunsten der VR-Umgebung bezüglich der Bearbeitungszeit der Explorations- und Analyseaufgaben. Wir vermuten, dass dieser Vorteil

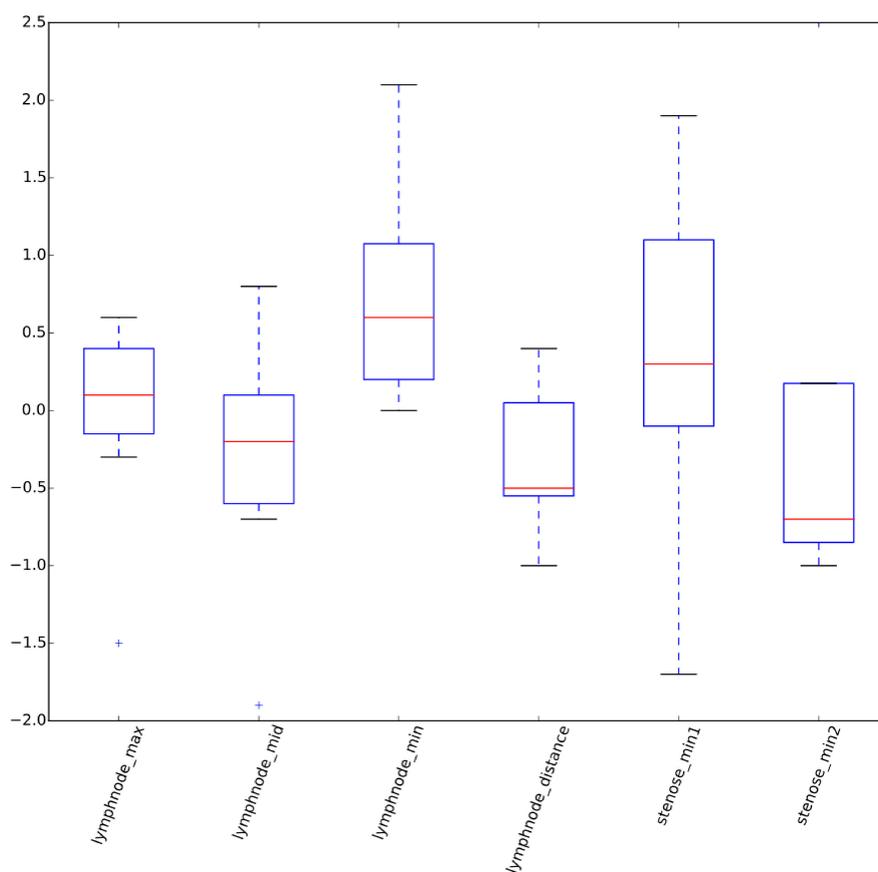


Abbildung 5.6: Unterschied zwischen den Messungen im 2D-System und den Messungen im VR-System. Die Messdaten des VR-Systems wurden von denen des 2D-Systems abgezogen. Alle Angaben sind in mm.

in der Bearbeitungsgeschwindigkeit in der deutlich direkteren, *natürlicheren* Eingabe liegt. Während die Probanden bei der Nutzung von Maus, Tastatur und 2D-Monitor gezwungen sind, mit Hilfe des eingeschränkten Aktionsraumes der Maus innerhalb eines eingeschränkten Sichtfeldes durch den flachen Monitor zu navigieren, ist bei der VR-Lösung nahezu das komplette Sichtfeld nutzbar und der Aktionsraum der frei beweglichen Controller nahezu nur durch die Bewegungsmöglichkeiten des Nutzers eingeschränkt.

Fragebogenauswertung

Nach dem Absolvieren aller Aufgaben, wurde von jedem Probanden ein Fragebogen ausgefüllt (siehe Anhang 1). Der Fragebogen teilt sich in zwei Fragegruppen auf. Die erste Fragegruppe bezieht sich nur auf die VR-Umgebung. Die zweite Gruppe zielt auf einen Vergleich des 2D-Systems und des VR-Systems ab. Der Fragebogen ist in Anhang 1 zu finden.

Wie in [Abbildung 5.8](#) zu sehen ist, sind die befragten Personen mit der bisherigen Implementierung zufrieden. Der Eindruck der 3D-Szene ist durch die VR-Umgebung gut gegeben und die Bedienung macht den Probanden Spaß. Die Funktionen der Tools verhalten sich wie gewünscht. Alle Probanden fanden die Clipping-Ebene als

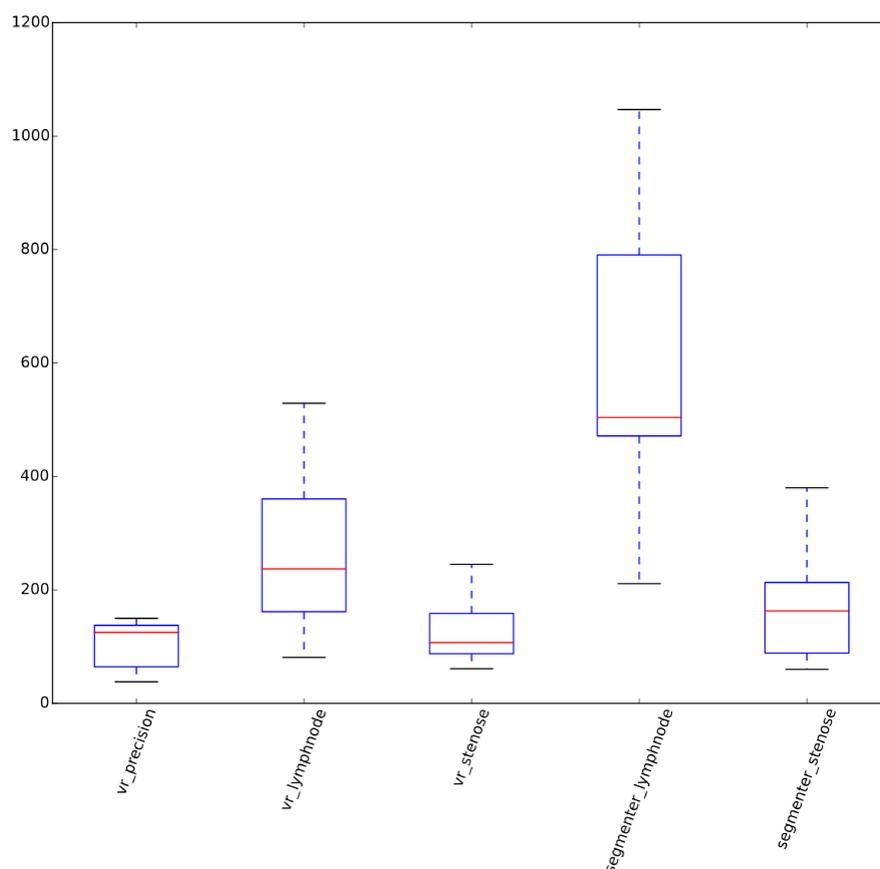


Abbildung 5.7: Zeit in Sekunden, die für jede Aufgabe benötigt wurde

hilfreich um die Aufgaben zu bewältigen und konnten sich auch vorstellen Segmentierungen in VR anzufertigen. Die Toolauswahl scheint noch verbesserungswürdig. Die Anstrengung beim Verwenden der VR-Umgebung scheint allerdings unkomfortabel zu sein. Zwei Probanden fanden dies akzeptabel, fünf empfanden es jedoch als ermüdend mit dem System zu arbeiten.

Beim zweiten Teil (Abbildung 5.9) der Fragen ist der Vergleich zwischen 2D-Software und VR-Software im Vordergrund. Auch hier ist zu erkennen, dass VR zumindest für den Körper ermüdender ist. Die mentale Anstrengung ist allerdings von Person zu Person anders. Dort sind sich die Probanden uneinig. Dies kann aber auch durch vorherige mentale Belastung, z.B. durch einen vorherigen Arbeitstag oder Ähnliches hervorgerufen, beeinflusst werden. In allen anderen Fragen lässt sich erkennen, dass das VR-System in Exploration und Handhabung der Tools in einem 3D-Raum Vorteile bietet. Laut der Probanden lassen sich hier relative Abstände besser einschätzen, Ansatzpunkte für die Messpfeile schneller auffindig machen und dann besser platzieren. Alles in allem scheinen die Probanden die Arbeit im VR-System als die präzisere Technologie einzuschätzen.

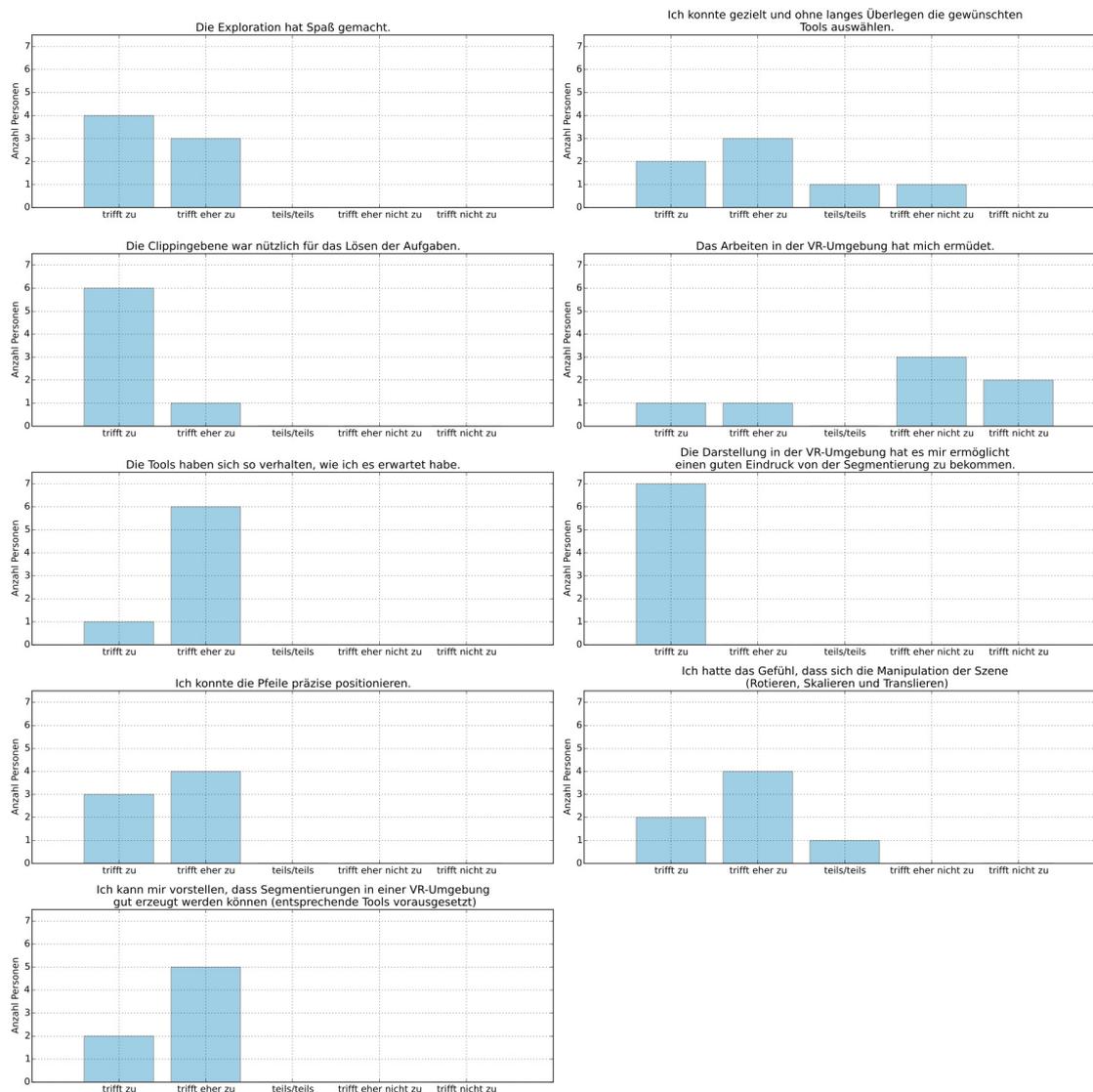


Abbildung 5.8: Fragen bezüglich der VR-Umgebung

Anmerkungen der Probanden

Nach den Tests hatten einige Probanden Anmerkungen zu unserer entwickelten VR-Software gegeben. Ein von Nutzern hervorgebrachter Kritikpunkt ist das Tracking der Controller, da mitunter Aussetzer des Trackings dazu geführt hatten, dass plötzlich die Controller nicht mehr an der erwarteten Stelle waren. Dies kann an ungünstigen Umwelteinflüssen liegen, die die Hardware beeinflussen. Bedingt durch die zur Verfügung stehenden Räumlichkeiten musste das VR-System in der Nähe von zur Sonnenseite ausgerichteten Fenstern aufgebaut werden, wodurch das Infrarotlicht der Sonne die Sensoren von HMD und Controllern beeinflussen konnte. Andere Bemerkungen bezogen sich auf die Nutzererfahrung bei Verwenden der Clippingplane. So wurde bemängelt, dass sich die Clippingplane nicht direkt mit der Triggertaste festsetzen lässt, sondern das Menü auf dem Touchpad dazu bedient werden muss. Zudem sei es wünschenswert mehrere Clippingplanes in der Szene platzieren zu können. Derzeit lässt sich nur eine Clippingplane setzen. Weiterhin wurde die Lesbarkeit

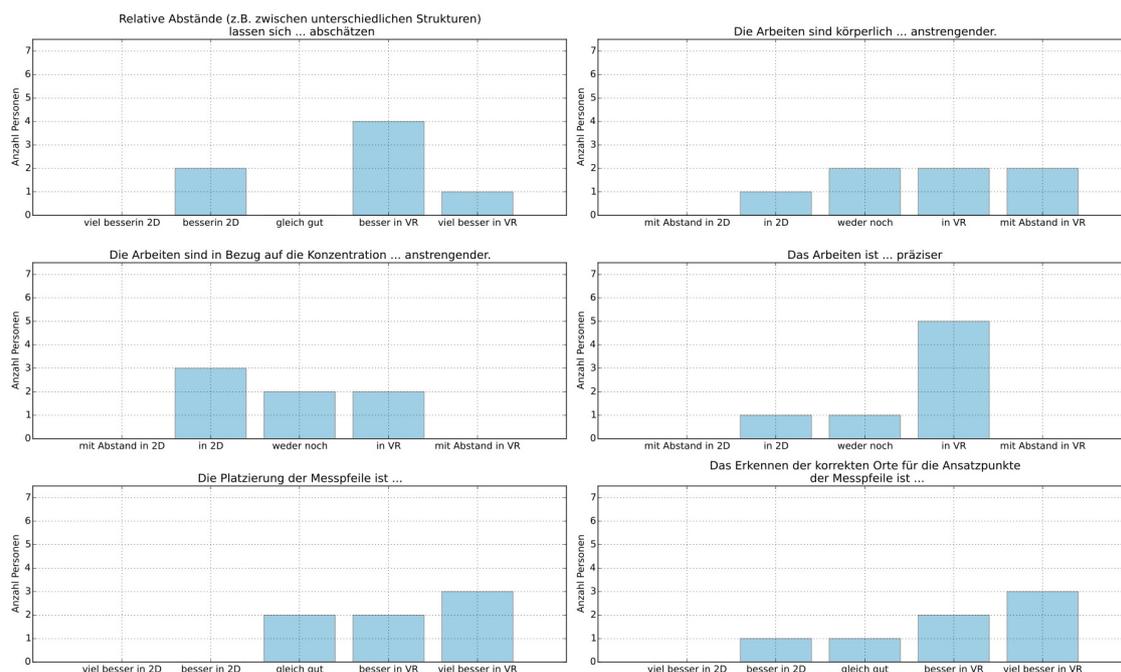


Abbildung 5.9: Fragen bezüglich des Vergleichs zwischen 2D-System und VR-Umgebung

der Schrift an den Pfeilen bemängelt, da diese zu wenig Kontrast hatten oder durch die begrenzte Auflösung der VR-Brille bei weit entfernten Messpfeilen nicht deutlich lesbar waren. Ein Proband merkte zusätzlich an, dass sich die Rotation während der Skalierung nicht so verhält, wie man es erwarten würde. Hier ist noch eine genauere Untersuchung nötig, ob es weitere sinnvolle Interaktionen für eine Rotation mit zwei Controllern gibt und was im Allgemeinen als angenehm empfunden wird.

6. Zusammenfassung

Das wissenschaftliche Teamprojekt behandelt den Vergleich aktueller VR-Technologie mit dem bisher üblichen Gebrauch von 2D-Monitoren, in Form einer Nutzerstudie. Um die Studie durchführen zu können, wurde zunächst eine Software entwickelt, welche die Exploration von medizinischen Datensätzen mittels VR-Hardware erlaubt. Die Hardware besteht aus einem Head-Mounted-Display und zwei frei beweglichen Controllern. Die erarbeitete VR-Software erlaubt mithilfe der Hardware Basisinteraktionen und einfache Messoperationen.

Die durchgeführte Nutzerstudie zeigt einen direkten Vergleich der konventionellen 2D-Software mit der von uns implementierten VR-Software. Diese zeigt, dass das VR-System in der Wahrnehmung der Datensätze Vorteile bietet. Der Datensatz wurde schneller exploriert und die Probanden hatten das Gefühl Messungen innerhalb des Datensatzes besser platzieren zu können. Der Vergleich zwischen den Vermessungsergebnissen zeigt, dass es Unterschiede zwischen den Messungen im 2D-System und im VR-System gibt. Dabei fällt auf, dass größere Strukturen größere Messunterschiede aufweisen, als kleinere Strukturen. Unsere Vermutung ist, dass im VR-System diagonale Messungen einfacher und präziser durchgeführt werden, falls diese nicht in einer achsenparallelen Ebene sind. Beim 2D-System wird meist eine Ebene eingeblendet und die Benutzer sind versucht, nur in dieser zu vermessen. Hier könnte eine ausgeweitete Nutzerstudie mit Einbeziehung von professionell ermittelten Messwerten, nochmals nachprüfen, ob VR-Systeme zu genaueren Vermessungen führen. Bei der Zeit, die die Probanden gebraucht haben, zeigt die Nutzerstudie, dass das VR-System wesentlich schnelleres Arbeiten ermöglicht. Im Durchschnitt wurden die Aufgaben in wesentlich geringerer Zeit gelöst. Dafür ist die Arbeit in VR körperlich anstrengender. Mental scheinen 2D- und VR-Umgebung eine subjektiv vergleichbare Belastung zu erzeugen.

Alles in allem scheint das VR-System einige Vorteile zu bieten, wie Zeitersparnis und das Gefühl präziserer Interaktionsmöglichkeiten. Die Vermessungsergebnisse unterscheiden sich zu den bisherigen 2D-Systemen. Ob die Vermessungen dadurch genauer oder ungenauer werden gilt es dagegen noch herauszufinden.

A. Anhang

A.1 Fragebogen

Die nachfolgenden Fragen beziehen sich ausschließlich auf die VR-Umgebung:

Die Exploration hat Spaß gemacht. Trifft:

völlig zu überwiegend zu teils/teils zu überwiegend nicht zu überhaupt nicht zu

Ich konnte gezielt und ohne langes Überlegen die gewünschten Tools auswählen. Trifft:

völlig zu überwiegend zu teils/teils zu überwiegend nicht zu überhaupt nicht zu

Die Clippingebene war nützlich für das Lösen der Aufgaben. Trifft:

völlig zu überwiegend zu teils/teils zu überwiegend nicht zu überhaupt nicht zu

Das Arbeiten in der VR-Umgebung hat mich ermüdet. Trifft:

völlig zu überwiegend zu teils/teils zu überwiegend nicht zu überhaupt nicht zu

Die Tools haben sich so verhalten, wie ich es erwartet habe. Trifft:

völlig zu überwiegend zu teils/teils zu überwiegend nicht zu überhaupt nicht zu

Die Darstellung in der VR-Umgebung hat es mir ermöglicht einen guten Eindruck von der Segmentierung zu bekommen. Trifft:

völlig zu überwiegend zu teils/teils zu überwiegend nicht zu überhaupt nicht zu

Ich konnte die Pfeile präzise positionieren. Trifft:

völlig zu überwiegend zu teils/teils zu überwiegend nicht zu überhaupt nicht zu

Ich hatte das Gefühl, dass sich die Manipulation der Szene (Rotieren, Skalieren und Translieren) entsprechend meiner Erwartung verhielt. Trifft:

völlig zu überwiegend zu teils/teils zu überwiegend nicht zu überhaupt nicht zu

Ich kann mir vorstellen, dass Segmentierungen in einer VR-Umgebung gut erzeugt werden können (entsprechende Tools vorausgesetzt). Trifft:

völlig zu überwiegend zu teils/teils zu überwiegend nicht zu überhaupt nicht zu

Die folgenden Fragen beziehen sich auf einen Vergleich des 2D-Maus- und -Monitor-Setup (im Folgenden 2D) mit der VR-Umgebung (im Folgenden VR):

Relative Abstände (z.B. zwischen unterschiedlichen Strukturen) lassen sich ...

viel besser in 2D besser in 2D gleich gut besser in VR viel besser in VR
... abschätzen.

Die Arbeiten sind körperlich ...

mit Abstand in 2D in 2D weder noch in VR mit Abstand in VR
... anstrengender.

Die Arbeiten sind in Bezug auf die Konzentration ...

mit Abstand in 2D in 2D weder noch in VR mit Abstand in VR
... anstrengender.

Das Arbeiten ist ...

mit Abstand in 2D in 2D weder noch in VR mit Abstand in VR
... präziser.

Die Platzierung der Messpfeile ist ...

viel besser in 2D besser in 2D gleich gut besser in VR viel besser in VR

Das Erkennen der korrekten Orte für die Ansatzpunkte der Messpfeile ist ...

viel besser in 2D besser in 2D gleich gut besser in VR viel besser in VR

A.2 Interaktionsdaten

Interaktionsdaten

Testperson:	1	2	3	4	5	6	7
VR-Interaktionsaufgabe:							
Anzahl an Grabs	7	19	4	16	29	22	7
Anzahl an Skalierungen	2	3	1	12	8	4	2
Skalierungsfehler (% Abweichung)	1.39628	2.72253	3.7824	2.46095	2.43632	2.75885	4.26227
Rotationenfehler (in grad)	2.06745	1.254	1.42101	2.6013	2.40194	1.624	2.19254
Translationenfehler (in m)	0.042437	0.00403269	0.00707994	0.00544041	0.00653623	0.00427355	0.00596912
Lymphknoten Vermessung:							
Trigger benutzt	66	45	20	67	65	47	64
Touchpad benutzt	30	24	28	26	34	10	7
Grab benutzt	46	26	12	45	47	42	55
Skalierung benutzt	9	10	10	28	4	3	7
Clippingebene erzeugt	3	1	1	1	1	1	0
Clippingebene bewegt	3	1	2	1	4	1	0
Clippingebene festgesetzt	6	2	3	2	2	2	0
Clippingebene entfernt	3	0	0	0	0	0	0
Messpfeil hinzugefügt	7	10	5	12	7	5	7
Messpfeil versucht zu bewegen	1	0	0	0	1	0	1
Messpfeil bewegt	9	5	1	0	8	0	0
Messpfeil versucht zu entfernen	0	0	0	4	0	0	0
Messpfeil entfernt	3	4	2	8	2	0	1
Home Button benutzt	0	0	1	0	0	0	0
Stenose Vermessung:							
Trigger benutzt	14	15	19	32	41	39	59
Touchpad benutzt	7	4	16	14	9	9	4
Grab benutzt	13	13	11	30	33	32	57
Skalierung benutzt	4	5	3	28	6	2	14
Clippingebene erzeugt	1	1	1	2	1	1	1
Clippingebene bewegt	2	0	3	2	1	1	0
Clippingebene festgesetzt	3	1	4	4	2	2	1
Clippingebene entfernt	1	0	0	2	0	0	0
Messpfeil hinzugefügt	1	2	5	2	4	7	2
Messpfeil versucht zu bewegen	0	0	0	0	0	0	0
Messpfeil bewegt	0	0	0	0	0	0	0
Messpfeil versucht zu entfernen	0	0	1	0	0	0	0
Messpfeil entfernt	0	0	2	0	2	0	0
Home Button benutzt	0	0	0	0	0	0	0

A.3 Vermessungsergebnisse

Vermessungsergebnisse in mm

Testperson:	1	2	3	4	5	6	7
seg_lymphnode_max	5.5	5.5	5.6	5.7	5.3	4.3	5.5
vr_lymphnode_max	5.5	5.4	5.0	5.1	5.6	5.8	5.3
seg_lymphnode_mid	5.2	3.3	4.4	4.0	3.8	2.7	4.0
vr_lymphnode_mid	4.9	3.8	3.6	4.1	4.5	4.6	4.2
seg_lymphnode_min	3.2	3.0	2.9	3.2	3.0	2.0	3.0
vr_lymphnode_min	3.0	2.8	1.9	1.1	3.0	-	1.9
seg_lymphnode_distance	1.5	1.5	1.4	1.3	1.4	1.7	1.3
vr_lymphnode_distance	1.1	2.0	1.9	2.3	1.2	1.8	1.9
seg_stenose_min1	2.5	2.7	1.7	4.2	3.9	5.0	5.0
vr_stenose_min1	2.2	3.1	3.4	2.4	3.5	3.1	4.8
seg_stenose_min2	-	3.6	2.4	3.3	4.7	3.7	5.0
vr_stenose_min2	-	4.4	3.4	-	2.2	-	5.6

A.4 Bearbeitungszeiten

Bearbeitungszeiten in Sekunden

Testperson:	1	2	3	4	5	6	7
vr_precision	90	150	39	146	125	129	38
vr_lymphnode	529	196	237	378	343	127	81
vr_stenose	77	61	245	98	107	134	183
segmenter_lymphnode	563	1047	504	1018	493	211	450
segmenter_stenose	163	110	67	195	60	231	380

Literaturverzeichnis

- [BHS⁺14] Alexandra Baer, Antje Hübler, Patrick Saalfeld, Douglas W Cunningham, and Bernhard Preim. A comparative user study of a 2d and an autostereoscopic 3d display for a tympanoplastic surgery. In *VCBM*, pages 181–190, 2014. (zitiert auf Seite 2)
- [BS06] Reinhard Beichel and Dieter Schmalstieg. Liver surgery planning using virtual reality. 2006. (zitiert auf Seite 4)
- [CNSD93] Carolina Cruz-Neira, Daniel J Sandin, and Thomas A DeFanti. Surround-screen projection-based virtual reality: the design and implementation of the cave. In *Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, pages 135–142. ACM, 1993. (zitiert auf Seite 3)
- [GDPM08] Luigi Gallo, Giuseppe De Pietro, and Ivana Marra. 3d interaction with volumetric medical data: experiencing the wiimote. In *Proceedings of the 1st international conference on Ambient media and systems*, page 14. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2008. (zitiert auf Seite 4)
- [HWHK16] Claudia Hänel, Benjamin Weyers, Bernd Hentschel, and Torsten W Kuhlen. Visual quality adjustment for volume rendering in a head-tracked virtual environment. *IEEE transactions on visualization and computer graphics*, 22(4):1472–1481, 2016. (zitiert auf Seite 4)
- [MSG⁺14] Koosha Mirhosseini, Qi Sun, Krishna C Gurijala, Bireswar Laha, and Arie E Kaufman. Benefits of 3d immersion for virtual colonoscopy. In *3DVis (3DVis), 2014 IEEE VIS International Workshop on*, pages 75–79. IEEE, 2014. (zitiert auf Seite 3)
- [RCD⁺09] Ivo Rössling, Christian Cyrus, Lars Dornheim, Peter Hahn, Bernhard Preim, and Andreas Boehm. Interaktive Visualisierung von Abständen und Ausdehnungen anatomischer Strukturen für die Interventionsplanung. In *Bildverarbeitung für die Medizin 2009*, pages 381–385. Springer, 2009. (zitiert auf Seite 4)
- [RCD⁺10] Ivo Rössling, Christian Cyrus, Lars Dornheim, Andreas Boehm, and Bernhard Preim. Fast and flexible distance measures for treatment planning. *International journal of computer assisted radiology and surgery*, 5(6):633–646, 2010. (zitiert auf Seite 5)

- [RMB⁺15] David Randall, Peter Metherall, Karna Dev Bardhan, Paul Spencer, Richard Gillott, Rebecca de Noronha, and John W Fenner. The oculus rift virtual colonoscopy: introducing a new technology and initial impressions. *Journal of Biomedical Graphics and Computing*, 6(1):34, 2015. (zitiert auf Seite 3)
- [Sut68] Ivan E Sutherland. A head-mounted three dimensional display. In *Proceedings of the December 9-11, 1968, fall joint computer conference, part I*, pages 757–764. ACM, 1968. (zitiert auf Seite 2)
- [VBM⁺16] Jose E Venson, Jean Berni, Carlos S Maia, A Marques da Silva, Marcos d’Ornelas, and Anderson Maciel. Medical imaging VR: can immersive 3D aid in diagnosis? In *Proceedings of the 22nd ACM Conference on Virtual Reality Software and Technology*, pages 349–350. ACM, 2016. (zitiert auf Seite 3)