Otto-von-Guericke-Universität Magdeburg
Fakultät für Informatik
Institut für Simulation und Graphik

Bachelorarbeit

Räumliches Skizzieren vaskulärer Strukturen und Pathologien mit dem 3D User Interface zSpace

Aleksandar Stojnic

Betreuer:
Prof. Dr.-Ing. Bernhard Preim
Patrick Saalfeld, M. Sc.

Selbstständigkeitserklärung

Hiermit versichere ich, dass ich die vorliegende Bachelorarbeit selbstständig und nur unter Zuhilfenahme der angegebenen Quellen erstellt habe.

Magdeburg,_____ _____

Aleksandar Stojnic

Zusammenfassung

Diese Arbeit befasst sich mit dem räumlichen Skizzieren und Manipulieren von Blutgefäßen und Gefäßkrankheiten in 3D, um die ärztliche Aufklärung von Patienten über Krankheit und Krankheitsverlauf zu unterstützen. Es werden medizinische Grundlagen und Grundlagen zu Sketch-based Interfaces und 3D User Interfaces erarbeitet. Aus deren Erkenntnissen wird ein Prototyp konzeptioniert und implementiert. Die Blutgefäße werden durch implizite Oberflächen generiert. Für das 3D User Interface wird das zSpace verwendet, zu dessen Komponenten ein stereoskopisches Display, eine passive 3D-Brille mit Markern, die Bewegungsparallaxe ermöglichen und als Eingabegerät ein Stift mit sechs Freiheitsgraden (6-DOF) zählen. Der Prototyp wird qualitativ evaluiert, um zu überprüfen, wie sehr sich das zSpace zum Skizzieren und Manipulieren von Blutgefäßen und Gefäßkrankheiten eignet. Die Evaluierung ergab, dass alle Probanden simple Gefäßstrukturen, sowie Aneurysmen und Stenosen leicht zeichnen konnten.

Inhaltsverzeichnis

1 Einleitung

31% aller Todesfälle weltweit werden durch Gefäßkrankheiten verursacht [1]. Menschen, die von solchen Krankheiten betroffen sind, wissen oftmals nicht, welche Auswirkungen dies auf ihre Gesundheit haben könnte. Eine Aufklärung vom behandelnden Arzt zur Krankheit und zu deren Behandlungsmöglichkeiten kann hierbei eine Hilfe sein, Patienten ihre Situation zu erklären. Zusätzlich hat dies nach Keulers [2] weitere Vorteile für Patienten:

- Die Behandlung nimmt weniger Zeit in Anspruch,
- die Menge an Medikamenten kann reduziert werden,
- die Patienten gehen verantwortungsvoller mit ihrer Krankheit um und
- sie sind weniger auf den Rat des behandelnden Arztes angewiesen.

Skizzen dienen hierbei als eine Ergänzung zur Aufklärung. Sie präsentieren und kommunizieren Konzepte, indem diese durch Symbole und grafische Elemente abstrahiert werden [3]. Patienten können dadurch Informationen zur Krankheit, zum Krankheitsverlauf und zu Behandlungsmethoden vermittelt bekommen, ohne dass sie Vorkenntnisse über das medizinische Feld besitzen müssen. Zudem sind Skizzen einfach und schnell zu zeichnen.
Skizzen, die mit Stift und Papier gezeichnet werden, haben allerdings einige Nachteile [4] (siehe Abbildung 1):

- Zeichnungen werden unordentlicher durch Überzeichnungen und Korrekturen.
- Einzelne Elemente sind schwierig zu erkennen, da nur eine Farbe genutzt wird.
- Patienten haben Schwierigkeiten bei der Interpretation der Skizze.

Abbildung 1: Papierskizze eines Blutgefäßes, in der Krankheit und Behandlungsmöglichkeiten illustriert sind (Bildquelle: [4]).

Diese Probleme können durch Softwarelösungen umgangen werden. Von Saalfeld [5] wurde eine Anwendung entwickelt, mit der Blutgefäßstrukturen, Pathologien und Behandlungsmöglichkeiten skizziert werden können (siehe Abbildung 2). Sie bietet zudem die Möglichkeit, den Blutfluss zu simulieren. Skizzen mit dieser Anwendung sind übersichtlicher, da sie zum einen mehrere Farben verwenden. Zum anderen können Teile der Skizze ohne Überreste gelöscht werden. Darüber hinaus sind einzelne Objekte klar voneinander zu unterschieden.

Abbildung 2: Die von Saalfeld [5] entwickelte Anwendung zum Skizzieren von Blutgefäßen, Pathologien und Behandlungsmöglichkeiten mit integriertem Blutfluss. Dem Nutzer stehen links Werkzeuge zum Zeichnen und Löschen von Gefäßen (1, 2), Platzierung von Objekten für Behandlungen (3, 4, 5) und zur Spezifizierung des Blutflusses (6, 7) zur Verfügung. Zusätzlich hat der Nutzer die Möglichkeit, ein Bild als Hintergrund zu laden (8) und alles zu löschen (9). Rechts befinden sich Parameter zur Anpassung der Blutflusssimulation (Bildquelle: [5]).

Zwar wird durch eine 2D Repräsentation die Realität abstrahiert und vereinfacht, allerdings bedeutet das auch, dass die Skizze nicht wie ein echtes Blutgefäß aussieht. Zudem kann eine 2D Skizze nur einen Blickwinkel darstellen. Es können somit wichtige Informationen fehlen, wodurch es trotzdem noch zu Fehlinterpretationen seitens der Patienten kommen kann.
Deshalb ist das Ziel dieser Arbeit, das Konzept von Saalfeld [5] in 3D zu erweitern, wobei nur das interaktive Zeichnen von Blutgefäßen und Pathologien im 3D Raum umgesetzt wird. Das hat den Vorteil, dass Blutgefäße realer dargestellt werden und aus verschiedenen Blickwinkeln betrachtet werden können. Somit lassen sich mehr Informationen vermitteln und Fehlinterpretationen vermeiden.
Allerdings stellt dies höhere Anforderungen an die Interaktion zwischen Nutzer und Software. Zeichnen im Raum ist schwieriger mit klassischen Benutzungsschnittstellen umsetzbar, weshalb in dieser Arbeit das 3D User Interface zSpace verwendet wird. Dieser hat als einer seiner Komponenten einen Stift mit sechs Freiheitsgraden, mit dem räumliche Eingaben getätigt werden können. Dadurch wäre ein freies Zeichnen im Raum möglich. Ein Schwerpunkt dieser Arbeit wird sein zu überprüfen, ob die zur Implementierung verwendeten Eingabegeräte, Ausgabegeräte und Interaktionstechniken für das Zeichnen von Blutgefäßen und Pathologien geeignet sind.

Im zweiten Kapitel werden medizinische Grundlagen zu Blutgefäßen und Pathologien erläutert, sowie die verwendeten Benutzungsschnittstellen und implizite Oberflächen vorgestellt.
Im dritten Kapitel folgt die Konzeption eines Prototyps, der in Bezug auf das Zeichnen, Editieren

und Erstellen von Blutgefäßen und Pathologien erklärt wird.
Im vierten Kapitel wird auf die Implementierung des Konzeptes mit der Game Engine Unity und dem zSpace eingegangen.
Anschließend wird im fünften Kapitel die qualitative Evaluierung des Prototyps mit der Think-Aloud Methode und mit Fragebögen beschrieben und die Ergebnisse diskutiert.
Abschließend folgen im sechsten Kapitel eine Zusammenfassung und ein Ausblick darauf, wie der Prototyp verbessert und erweitert werden kann.

2 Grundlagen und Verwandte Arbeiten

In diesem Kapitel wird zuerst auf Aufbau von Blutgefäßen, Gefäßkrankheiten und auf Verlauf und Inhalt der Patientenaufklärung eingegangen. Anschließend folgt ein Überblick über Sketch-based Interfaces zur Modellierung und über 3D User Interfaces, wobei gleichzeitig verwandte Arbeiten beschrieben werden. Danach werden implizite Oberflächen vorgestellt und deren Eigenschaften, Arten und Darstellungsmöglichkeiten erklärt.

2.1 Medizinische Grundlagen

2.1.1 Blutgefäße und Gefäßerkrankungen

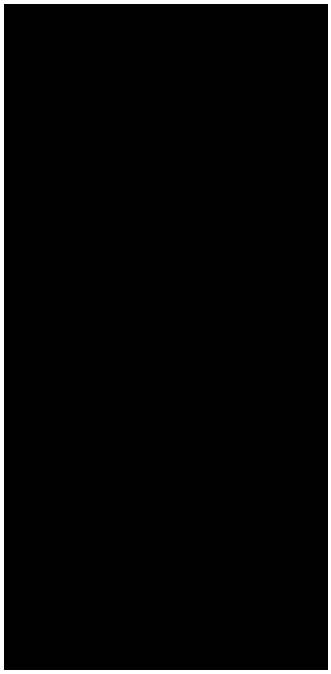


Abbildung 3: Darstellung des Blutgefäßsystems des Menschen. Arterien sind in rot und Venen in blau gefärbt (Bildquelle: nach jameda[1]).

Die folgenden Erklärungen zu Blutgefäßen und Gefäßerkrankungen basieren auf den Arbeiten von Zilles et al. [6] und Riede et al. [7].
Blutgefäße sind röhrenförmige Strukturen, die sich durch den gesamten Körper als eine komplexe verzweigte Baumstruktur erstrecken (siehe Abbildung 3). Ihre Aufgabe ist es, das Blut durch den Körper in die verschiedenen Organe zu transportieren.
Ein Blutgefäß besteht aus drei Schichten:

- **Tunica Intima:** Dies ist die innerste Schicht aus Endothelzellen. Ihre Funktion ist es, das Wachstum und die Entwicklung des Gefäßes zu kontrollieren. Zudem dient sie zur Blutgerinnung und zur Regulierung des Gefäßdurchmessers.

1 http://www.jameda.de/krankheiten-lexikon/bilder/big/560023.jpg, zuletzt aufgerufen: 08.02.2016

- **Tunica Media:** Dies ist eine Schicht aus glatten Muskelzellen, Kollagenfasern und elastischen Netzen. Sie steuert die Weite sowie die Steifheit des Gefäßes.
- **Tunica Externa:** Diese Schicht befindet sich außen und besteht aus Bindegewebe. Sie sorgt für die Platzierung des Gefäßes im Körper.

Blutgefäße lassen sich zudem grob unterteilen in:

- Venen, die Blut in Richtung zum Herzen transportieren,
- Arterien, die Blut in entgegengesetzte Richtung zum Herzen transportieren und
- Kapillaren, die Venen und Arterien miteinander verbinden.

Die Hauptursache von vielen Gefäßkrankheiten ist Arteriosklerose. Hierbei lagern sich Kalk, fibröses Gewebe und andere Ablagerungen zwischen der Tunica media und Tunica externa an.
Das kann zum einen zu einer Verengung der Blutgefäßwand führen, was als *Stenose* bezeichnet wird (siehe Abbildung 4). Dadurch können sich Thromben bilden oder die Blutzufuhr in wichtige Bereiche des Körpers unterbunden werden.

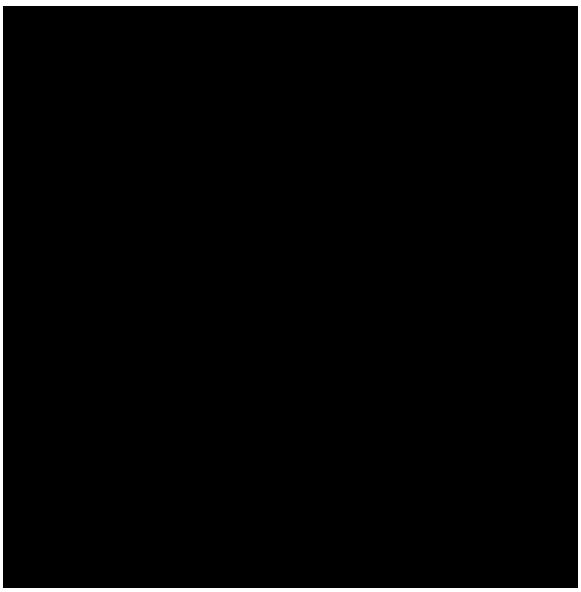


Abbildung 4: Darstellung einer durch Arteriosklerose verursachten Stenose mit daraus folgender Reduzierung des Blutflusses (A) und Thrombenbildung (B) (Bildquelle: mayfieldclinic[2]).

Zum anderen kann eine Arteriosklerose zu einem Verlust der Elastizität der Gefäßwand führen, wodurch sich sack- oder spindelförmige Erweiterungen des Blutgefäßes bilden. Diese werden als *Aneurysma* bezeichnet (siehe Abbildung 5). Platzt ein solches Aneurysma, dann entstehen innere Blutungen, die lebensgefährlich sein können.

2 http://www.mayfieldclinic.com/Images/PE-CarotidStenosis2.jpg, zuletzt aufgerufen: 08.02.2016

Abbildung 5: Darstellung eines sackfömigen (links) und spindelförmigen (rechts) Aneurysmas (Bildquelle: prweb[3]).

2.1.2 Patientenaufklärung

Nach Standl und Lussi [8] müssen Patienten von einem Arzt aufgeklärt werden, der Kenntnisse über das relevante Fachgebiet verfügt, wobei das nicht unbedingt der behandelnde Arzt sein muss. Zudem muss die Aufklärung so früh wie möglich erfolgen, damit Patienten genug Zeit haben zu überlegen, ob sie dem Eingriff zustimmen.
Die Aufklärung hat folgenden Inhalt [8]:

- Allgemeine Informationen zum Eingriff, wobei auch Nebeneingriffe eingeschlossen sind,
- Nebenwirkungen und Risiken,
- Chancen auf Heilung,
- gesundheitliche Folgen, falls kein Eingriff erfolgt und
- Alternativen bei besonderen Eingriffen.

Der Arzt vermittelt dem Patienten diese Informationen in einem persönlichen Gespräch [8]. Zusätzlich bekommen Patienten Aufklärungs- und Anamnesebögen, die im Voraus formuliert wurden, um einen Überblick zu geben. Skizzen und andere Hilfsmittel können als Ergänzung hinzugezogen werden. Ob diese Hilfsmittel verwendet werden, kann je nach medizinischem Bereich und Arzt unterschiedlich sein.

2.2 Benutzungsschnittstellen

2.2.1 Sketch-Based Interfaces zur Modellierung

Beginnend werden Sketch-based Interfaces (SBI) beschrieben, da diese die Grundlage für Sketch-based Interfaces zur Modellierung (SBIM) bilden.
Bei Sketch-based Interfaces handelt es sich um Benutzungsschnittstellen, deren Eingaben freihand mit einem direkten Zeigegerät (Stylus) geschehen [5]. Die Eingaben erfolgen entweder auf einem Tablet-PC oder auf einem Grafiktablett, dessen Ergebnis auf einem separaten Bildschirm angezeigt wird. SBIs nutzen direkte Manipulation als Interaktion, zu denen Techniken wie Selektion und Drag&Drop zählen [9]. Zusätzlich kann das Zeichnen mit dem Stylus genutzt werden, um Operationen auf Gesten abzubilden [10].
Mit SBIs ist es möglich, Ideen und Konzepte zu kommunizieren, ohne präzise zeichnen zu müssen

3 http://ww1.prweb.com/prfiles/2011/10/24/8903041/aneurysma.jpg, zuletzt aufgerufen: 08.02.2016

[11]. Zudem bevorzugen Nutzer nach Xu et al. [12] die Arbeit mit SBIs über die Arbeit mit klassischen Benutzungsschnittstellen. Ein Nachteil ist allerdings, dass es zu unerwünschten Ergebnissen kommen kann, wenn die Eingaben zu ungenau sind [13].
SBIs werden häufig zum Erstellen von 3D Modellen verwendet. Solche Anwendungen werden dementsprechend als *Sketch-based Interfaces zur Modellierung* (engl. Sketch-based Interfaces for Modeling) bezeichnet. Sie bieten eine Alternative zu klassischen Modellieranwendungen, deren Benutzungsschnittstellen sehr komplex und dadurch mit einem hohen Lernaufwand verbunden sind [14].

Abbildung 6: Pipeline einer SBIM-Anwendung (Bildquelle: [10]).

Olsen et al. [10] beschreiben, wie die Pipeline einer typischen SBIM-Anwendung aussieht (siehe Abbildung 6). Zuerst zeichnet der Nutzer die Skizze. Diese wird dann von der Anwendung bearbeitet und in eine für die Interpretation geeignetere Form gebracht. Anschließend wird die Skizze interpretiert und daraus ein 3D Modell erstellt.
Eine der ersten Anwendungen auf diesem Gebiet war *Teddy* [15]. Hier zeichnet der Nutzer eine geschlossene Kontur. Daraufhin folgt eine Delauny-Triangulierung, um dessen Oberflächenmodell (engl. Mesh) zu bestimmen. Anschließend werden die Mitten der Dreieckskanten, die nicht zur Kontur gehören, miteinander verbunden, wodurch eine Mittellinie im Inneren entsteht. Diese Mittellinie wird als das *Skelett* der Kontur bezeichnet. Das Skelett wird dann in Richtung der z-Achse angehoben und deren Eckpunkte (engl. Vertices) mit den Eckpunkten der Kontur verbunden. Zuletzt wird das daraus entstandene Modell an der z-Achse gespiegelt, wodurch eine runde Form als Endergebnis entsteht. Dieser Prozess wird *Inflation* genannt, da die Kontur „aufgeblasen“ wird, um es in ein 3D Modell umzuwandeln (siehe Abbildung 7 (a)). Das Modell kann anschließend rotiert werden, um es aus verschiedenen Blickwinkeln zu betrachten.
Teddy bietet zudem die Möglichkeit, per Extrusion neue Teilmodelle an das initiale Modell anzufügen (siehe Abbildung 7 (b)). Dafür zeichnet der Nutzer einen Ring an das Modell und einen Extrusionsstrich. Der Ring wird dann entlang dieses Striches kopiert und die Kopien mit Dreiecken verbunden.
Als Editieroperationen gibt es *Schneiden*, *Malen* und *Glättung*. Für das *Schneiden* zeichnet der Nutzer einen Strich durch das Modell hindurch und alles, was sich links vom Strich befindet, wird gelöscht. Beim *Malen* wird die gezeichnete Linie auf das Modell projiziert und beim *Glätten* werden Beulen innerhalb eines gezeichneten Ringes entfernt.

Abbildung 7: Darstellung des Inlation- (a) und Extrusionsprozesses (b) von Teddy [15]. Bei Inflation wird das Skelett nach oben gehoben und dessesn Vertices mit den Vertices der Kontur verbunden. Bei der Extrusion wird der rote Ring entlang der gezeichneten Kontur kopiert. Die Kopien werden anschließend durch Dreiecke miteinander verbunden (Bildquelle: nach [15]).

Teddy wäre ein geeigneter Ansatz zum Zeichnen von Blutgefäßen, da hier das Skelett nicht bestimmt werden muss. Es würde ausreichen, nur eine Linie zu zeichnen, die das Skelett des Gefäßes festlegt. Per Extrusion könnte der Nutzer dann Baumstrukturen zeichnen.
Allerdings hat Teddys Ansatz viele Limitationen. Zum einen können nicht zwei geschlossene Konturen auf einmal gezeichnet werden, um daraus zwei Modelle zu erstellen. Zum anderen fehlen Operationen, um zwei oder mehr unterschiedliche Objekte zu kombinieren. Dadurch wäre es nicht möglich, zwei Blutgefäße miteinander zu verbinden. Darüber hinaus ist das Mesh, dass bei Inflation entsteht, von geringer Qualität, weshalb Teddy dieses nach dem Prozess optimiert. Das erfordert allerdings zusätzliche Berechnungszeit. Ein weiteres Problem ist, dass es bei der Extrusion zu viele Sonderfälle für das Löschen von Dreiecken gibt, wodurch die Implementierung sehr komplex werden könnte. Deshalb wird in dieser Arbeit auf Teddys Ansatz verzichtet.
Viele alternative Anwendungen nutzen eine Zwischenrepräsentation des Modells, um die Probleme von Teddy zu umgehen. Cherlin et al. [16] verwenden reguläre Flächen (engl. Parametric Surfaces), bei der die Fläche von Funktionen mit zwei Parametern beschrieben wird [17]. Röhrenförmige Modelle werden hierbei definiert durch zwei Kurven, die um eine Mittellinie rotiert werden (siehe Abbildung 8). Es ist damit möglich, simple Meshes zu erstellen, allerdings besteht weiterhin das Problem der vielen Sonderfälle bei Verästelungen.

Abbildung 8: Darstellung der Rotation zweier Kurven um eine Mittellinie. Die beiden Kurven $g_l(u)$ und $q_r(u)$ werden hierbei entlang des Kreises um c(u) in v Intervallen kopiert (Bildquelle: [16]).

Einige Anwendungen benutzen implizite Oberflächen als Zwischenrepräsentation. Karpenko et al. [18] definieren Einschränkungspunkte an den Vertices der gezeichneten Kontur (siehe Abbildung

9). Diese bilden ein lineares Gleichungssystem, dessen Lösung die Oberfläche definiert. Daraus wird dann das Mesh generiert. Hier ist es möglich, zwei Modelle miteinander zu verschmelzen, indem die Einschränkungspunkte der beiden zusammengefasst werden. Anschließend werden die Einschränkungspunkte entfernt, die sich innerhalb der Überlagerung der beiden Modelle befinden. Der Vorteil dieser Methode ist, dass stets glatte Oberflächen entstehen, allerdings wird das Lösen des Gleichungssystems rechenintensiver, je mehr Einschränkungspunkte es gibt.

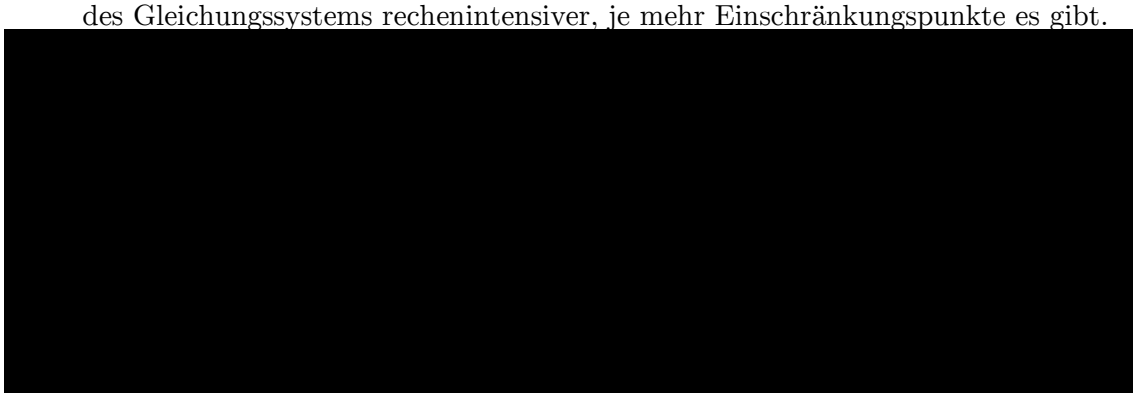


Abbildung 9: Einschränkungspunkte der Kontur. Die erste Menge an Einschränkungspunkten (Kreise) wird an den Vertices der Kontur definiert. Die Punkte der zweiten Menge (Plus-Zeichen) sind entlang der Normalen der ersten Menge verschoben. (Bildquelle: nach [18])

Eine alternative Nutzung impliziter Oberflächen wird von Alexe et al. [14] präsentiert. Hier wird zuerst eine Kontur gezeichnet und daraus das Skelett generiert, so wie bei Teddy. Danach werden sphärische implizite Oberflächen an den Vertices des Skeletts generiert. Der Radius der Flächen wird daraufhin erhöht, bis sie sich miteinander verschmelzen und die Kontur ausfüllen. Der Nutzer kann dann den Radius der einzelnen Flächen beeinflussen, um bestimmte Bereiche des Modells dicker oder dünner zu machen. Die Oberfläche des Modells bleibt dabei glatt. Dies zusammen mit der Verschmelzungseigenschaft würde es leicht möglich machen, Blutgefäße miteinander zu verbinden, sowie Stenosen und Aneurysmen nachzubilden. Deshalb werden in dieser Arbeit zur Generierung des Meshes implizite Oberflächen verwendet (siehe Abschnitt 3.3.1).

2.2.2 3D User Interfaces

Die folgenden Erklärungen zu 3D User Interfaces (3D UI) und deren Ein- und Ausgabegeräte basieren auf der Arbeit von Bowman et al. [19].
Bei 3D UIs handelt es sich um Benutzungsschnittstellen, bei denen der Nutzer mit dem Computer direkt im 3D-Raum interagiert. Solche Interaktionen eignen sich für Anwendungen, deren Aufgaben in einer 3D Umgebung stattfinden, da Aktionen sich an den erlernten Fähigkeiten des Nutzers in der realen Welt orientieren können. Zum Beispiel ist das Ausführen einer Schlagbewegung mit der Hand, um einen virtuellen Charakter schlagen zu lassen, natürlicher, als wenn nur ein Knopf dafür gedrückt wird. 3D UIs können dann möglichst reales Feedback auf solche Aktionen geben.
Diese Eigenschaften können z. B. zum Training für Militäreinsätze genutzt werden. Es ist zudem auch möglich, 3D Umgebungen für Simulationen und für die Visualisierung von komplexen 3D-Datensätzen zu nutzen. 3D UIs finden auch in Computerspielen Anwendung, bei denen sie dem Spieler erlauben, direkt auf die Spielwelt einzuwirken und somit Immersion zu erzeugen.

3D ist allerdings nur dann effektiv, wenn der Nutzer räumliche Tiefe wahrnehmen kann [19]. Deshalb müssen Ausgabegeräte entsprechende Hinweise zur visuellen Tiefe vermitteln. Es gibt hierbei verschiedene Arten:

Monokulare Hinweise
Tiefeninformationen werden durch 2D Bilder vermittelt, die mit nur einem Auge sichtbar sind. Zum Beispiel werden kleine Objekte in Bezug zu großen Objekten als weiter weg empfunden. Weitere Beispiele sind Verdeckungen, lineare Perspektive, und Licht- und Schatteneffekte.

Bewegungsparallaxe
Hier erfolgt die Bewegung des Objektes in Relation zum Betrachter. Objekte, die sich nah am Nutzer befinden, bewegen sich schneller als weit entfernte Objekte.

Binokulare Disparität
Dies beschreibt die Unterschiede zwischen den Bildern des rechten und linken Auges. Werden diese beiden Bilder miteinander kombiniert, entsteht ein Tiefeneffekt, der als *Stereoskopie* bezeichnet wird. Hierfür sind allerdings spezielle Displays notwendig.

Bowman et. al [20] unterteilen Displays in semi-immersiv und immersiv.
Zu semi-immersiven Displays zählen Stereo-Bildschirme. Diese nutzen Stereoskopie, um einen virtuellen Raum in der realen Welt zu definieren. Damit dies funktioniert, kann neben einer hohen Bildwiederholungsrate eine spezielle Bildschirmtechnologie verwendet werden. Solche Arten von Stereo-Bildschirmen werden als autostereoskopische Displays bezeichnet. Alternativ kann statt der Verwendung einer Bildschirmtechnologie eine spezielle Brille getragen werden. Es gibt hierbei zwei Arten, aktive und passive Brillen. Aktive Brillen, wie z. B. Shutter-Brillen, besitzen Verschlussklappen, die die Sicht eines einzelnen Auges jeweils abwechselnd verdecken. Die Rate, in der dies geschieht, entspricht der Bildwiederholungsrate. Damit diese beiden Raten synchron bleiben, werden Infrarot-Signale an die Brille gesendet.
Passive Brillen filtern zwei Bilder, die sich überlagern, durch entgegengesetzt polarisierte Filter an den Gläsern. Dadurch sieht der Nutzer auf einem Auge nur das rechte und auf dem anderen Auge nur das linke Bild. Die Stereoskopie bei solchen Brillen hat weniger Qualität als bei aktiven Brillen, da bei aktiven stets mit der Bildwiederholungsrate des Bildschirms synchronisiert wird. Allerdings kann die Qualität stark abnehmen, wenn z. B. die Brille durch Verdeckungen keine Infrarot-Signale für die Synchronisation empfangen kann. Bei passiven Brillen können solche Probleme nicht auftreten.
Zu immersiven Displays zählen Head Mounted Displays (HMD). Dies sind Geräte mit ein oder zwei Bildschirmen, die auf den Kopf des Nutzers aufgesetzt werden. Dadurch befinden sich die Bildschirme direkt vor den Augen. Ein Beispiel eines HMD ist die Occulus Rift (siehe Abbildung 10). Stereoskopie wird erreicht, indem auf jedem Bildschirm jeweils ein separates Bild dargestellt wird. Zudem kann Bewegungsparallaxe genutzt werden, indem der Nutzer seinen Kopf mit dem Display bewegt. Sie bekommen dadurch den Eindruck, dass sie sich innerhalb der virtuellen Welt befinden. Es ist allerdings eine hohe Auflösung erforderlich, da sich die Bildschirme sehr nah an den Augen befinden. Ansonsten kann das Pixelgitter sichtbar sein.

Abbildung 10: Darstellung von Occulus Rifts Head Mounted Display (Bildquelle: dbvc4uanumi2d.cloudfront[4]).

Damit nun der Nutzer mit der 3D Umgebung interagieren kann, ist ein Eingabegerät erforderlich. Eine wichtige Charakteristik hierbei ist die Anzahl der Freiheitsgrade (DOF). Diese beschreiben, wie sich ein Körper im Raum bewegt. Dazu zählen die Position und die Orientierung, aus denen sich insgesamt sechs Freiheitsgrade (6-DOF) ergeben.
Zudem können Eingabegeräte in aktive und passive unterteilt werden. Bei aktiven Geräten wird eine Eingabe erst ausgeführt, wenn der Nutzer physikalisch mit dem Gerät interagiert, z. B. beim Drücken eines Knopfes. Bei passiven Geräten geschieht eine Eingabe, ohne das der Einfluss des Nutzers nötig ist.
Zu aktiven Geräten zählen Tastatur, 2D Mäuse, Trackballs, Joysticks und Stift-basierte Tabletts. Diese sind für 2D-Desktop-Applikationen gedacht, jedoch können sie auch für 3D UIs genutzt werden. Allerdings besitzen die Geräte meist nur zwei DOF, weshalb Interaktionen limitiert sind. Es ist aber möglich, diese auf 6-DOF zu erweitern. Dafür muss die Bewegung des Gerätes durch *Motion Tracking* verfolgt werden.
Beim Motion Tracking wird die Position und die Orientierung eines physikalischen Objektes kontinuierlich abgefragt [21]. Diese Informationen können zur Beeinflussung von virtuellen Objekten genutzt werden. Eine Möglichkeit für Motion Tracking bietet *optisches Tracking*. Hier wird eine Lichtquelle genutzt, dessen Position und Lichtstrahlen von Kameras eingefangen wird. Alternativ können Sensoren verwendet werden, die diese Position und Orientierung mit Hilfe von Magnetfeldern ermitteln. Dies wird als *magnetisches Tracking* bezeichnet. Für weitere Methoden sei auf Bowman et. al [19] verwiesen.
Für 3D UIs gibt es folgende Gruppen von Interaktionstechniken [20]:

Navigation

Navigation beinhaltet Bewegung, Suchen und Manövrieren des Objektes durch die Umgebung. Je nachdem, welches Eingabe- und Ausgabegerät verwendet wird, kann dies auf unterschiedliche Arten erfolgen. Entweder wird das Gerät direkt bewegt oder der Nutzer führt eine Zieh-Bewegung

4 https://dbvc4uanumi2d.cloudfront.net/cdn/4.5.23/wp-content/themes/oculus/img/order/dk2-product.jpg, zuletzt aufgerufen: 11.02.2016

mit den Händen oder sonstige Bewegungen des Körpers aus. Andere Möglichkeiten sind z. B. per Steuerknüppel die Bewegungsrichtung anzugeben, oder ein Ziel auszusuchen, auf das sich automatisch bewegt werden soll.

Selektion und Manipulation
Hierfür gibt es zwei Möglichkeiten. Bei der einen wird ein virtueller Cursor verwendet, mit dem Objekte gegriffen, verschoben, rotiert und andere Aktionen ausgeführt werden können. Ein Problem ist allerdings, dass der Nutzer nur die Objekte manipulieren kann, die sich in Reichweite des Cursors befinden. Bei der anderen Möglichkeit wird ein Strahl ausgehend vom Eingabegerät in die Umgebung geschossen. Objekte, die sich mit dem Strahl schneiden, können selektiert und manipuliert werden.

Systemkontrolle
Hiermit sind Kommandos oder Aktionen gemeint, mit denen zwischen Zuständen in der Software gewechselt werden können. In klassischen Anwendungen erfolgt dies durch Elemente wie Buttons. Diese können aber nicht für 3D UIs verwendet werden, da es möglich ist, versehentlich durch die Elemente durchzugehen und sie zu verfehlen. Alternativ können dafür Stimmen- und Gesteneingaben, sowie virtuelle 3D Objekte genutzt werden.

2.2.3 Zusammenführung von Sketch-Based Interfaces und 3D User Interfaces

Wie bereits beschrieben wurde, haben SBIMs die Limitation, dass zuerst eine 2D Kontur des Modells gezeichnet werden muss. Mit 3D UIs wäre es möglich, Modelle direkt in 3D zu spezifizieren. Im folgenden werden drei Ansätze zur Zusammenführung von SBIs und 3D UIs beschrieben.
Perkunder et. al [22] verwenden ein existierendes SBIM Tool namens *FiberMesh* [23], der für 3D Eingaben angepasst wurde. Der Nutzer hat in der einen Hand einen Stylus, um die Kontur zu zeichnen und in der anderen eine Zange, mit der er die Linie greifen und verändern kann. Die Eingabegeräte wurden magnetisch getrackt. Als Ausgabegerät wird ein Surround-Screen Display verwendet, bei der eine Person zwischen drei oder mehreren Projektionsbildschirmen steht [19]. Zusätzlich trägt der Nutzer eine aktive 3D-Brille. Es wird zwar in 3D gezeichnet, allerdings wird die Eingabe wieder in 2D umgerechnet, da der FiberMesh Algorithmus ursprünglich nur 2D Eingaben akzeptiert. Somit ist es nicht möglich, Linien in die Tiefe zu zeichnen.

Abbildung 11: Zeichnen eines 3D Modells mit dem angepassten FiberMesh Algorithmus. In (a) wird per Stylus die Kontur gezeichnet. In (b) wird per Zange die Kontur gegriffen und durch Ziehen verändert (Bildquelle: nach [22]).

Laundry et al. [24] verwenden zwei Nintendo Wii Remotes, mit deren Infrarot-Kameras eine LED am Finger getrackt wird (siehe Abbildung 12 (a)). Dabei wurden verschiedene Wege getestet, wie Klickaktionen ausgelöst werden konnten, z. B. durch Verdecken einer zweiten LED oder durch Drücken eines Knopfes an einem Stylus. Als Ausgabegerät wurde ein großes Display verwendet. Die Eingaben mit diesem Aufbau waren allerdings nur präzise genug, um 3D Objekte zu verschieben, zu rotieren und zu skalieren.
Chen et al. [25] boten einen alternativen Aufbau für das Zeichnen in 3D (siehe Abbildung 12 (b)). Als Eingabegerät wurde ein 6-DOF Stylus verwendet, dessen Spitze für das Tracking farblich markiert war. Zusätzlich wurde ein Keyboard für Befehle genutzt. Tracking erfolgte mit einer Kamera und einem Spiegel, die links und rechts vom Nutzer platziert wurden. Das Ergebnis wurde dann auf einem normalen Computerbildschirm ausgegeben. Der Vorteil hier ist, dass die Umgebung einfacher zu kalibrieren ist, da es nur eine Kamera gibt. Dies vereinfacht das Tracking, da die Kamera den physikalischen Stylus und dessen Spiegelbild einfängt, wodurch eine zweite Kamera überflüssig ist. Das erlaubt insgesamt eine größere 3D Arbeitsfläche.

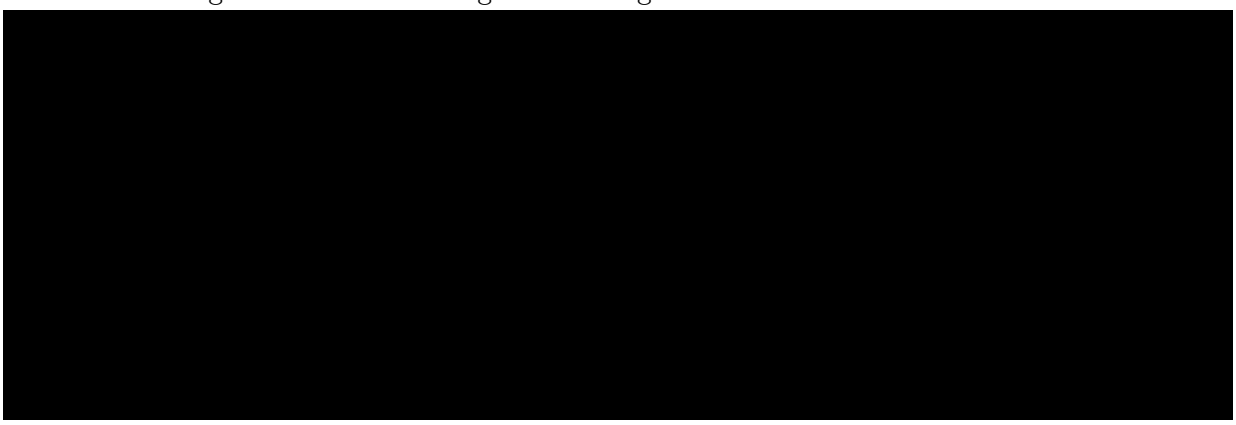


Abbildung 12: Aufbau der Umgebung von Laundry et al. (a) und Chen et. Al (b) (Bildquelle (a): [24], Bildquelle (b): [25]).

Die letzten beiden Ansätze haben eines gemeinsam. Die gezeichnete Linie in 3D ist am Stylus nicht zu sehen. Der Nutzer kann diese nur am großen Display oder am Bildschirm betrachten. Dadurch

kann es schwierig sein, die genaue Position einer Linie zu finden.

2.2.4 Usability Evaluierung von 3D User Interfaces

Usability beschreibt, wie sehr eine Software einem Nutzer dazu verhelfen kann, seine Ziele so effektiv und effizient wie möglich zu erreichen [26].
Klassische Benutzungsschnittstellen bieten viele Möglichkeiten, Software nach einheitlichen Standards zu evaluieren. Jedoch kann ein Großteil dieser Möglichkeiten nur schwer auf 3D UIs übertragen werden. Bowman et al. [19] schildern Probleme, die es bei der Evaluierung von 3D UIs gibt.
Zum einen unterscheidet sich die physikalische Umgebung von 3D UIs sehr von der Umgebung klassischer Benutzungsschnittstellen. 3D UIs haben oft ungewohnte Eingabe-, Ausgabegeräte und Interaktionskonzepte. Zum Beispiel kann es sein, dass der Nutzer sich durch einen Raum bewegt, weshalb im Vergleich zu einem Desktop-PC nach völlig anderen Gesichtspunkten evaluiert werden muss.
Das Problem hierbei ist, dass es keine Standards für 3D UI Design gibt. Deshalb sind Evaluierungen meist heuristisch und werden nach gröberen Gesichtspunkten untersucht, z. B. ob die verwendeten Ein- und Ausgabegeräte überhaupt geeignet sind.
Durch die andere physikalische Umgebung kann zudem die Evaluierung sehr komplex werden, da 3D UIs multimodale Eingaben verwenden können, wie z. B. Gesten, Stimme, oder Bewegungen [19]. Deshalb muss in solchen Situationen die Evaluierung von mehreren Personen durchgeführt werden, die jeweils einen Teil der Eingabe verfolgen.
Es ist aber dennoch möglich, etablierte Methoden für Evaluierungen auf 3D UIs anzuwenden. Sie müssen lediglich auf deren Besonderheiten angepasst werden. Folgend sind einige Methoden aufgeführt:

Cognitive Walkthrough [27]
Hier führen Experten die einzelnen Aktionen der Benutzungsschnittstelle aus und evaluieren, wie einfach es ist, ohne Hilfe von Anweisungen, Aufgaben zu erledigen. Im Fokus steht hierbei, wie schnell die Nutzer lernen, mit der Benutzungsschnittstelle umzugehen. Ein Cognitive Walkthrough erfolgt in den frühen Phasen des Designs.

Heuristische Evaluierung [28]
Experten bewerten hier die Benutzungsschnittstelle nach bestimmten Regeln und Design-Richtlinien. Es werden aber nur die Richtlinien verwendet, die auf 3D UIs anwendbar sind [19]. Es soll hierbei darauf aufmerksam gemacht werden, was an der Benutzungsschnittstelle gut ist und was noch verbessert werden muss.

Formative Evaluierung [19]
Der Nutzer wird in aufgaben-basierte Szenarios gebracht, um Usability-Probleme zu identifizieren. Das Design wird zudem dabei bewertet, ob es Erforschung, Lernen und Aufgabenperformanz unterstützt. Nachdem die daraus resultierenden Verbesserungen in das 3D UI eingebaut worden sind, wird der Prozess wiederholt.

Summative Evaluierung [29]
Die Evaluierung erfolgt hier kurz vor der Fertigstellung der Software. Es werden dabei alle Aspekte bewertet, um noch übrig gebliebene Probleme zu beseitigen.

2.3 Implizite Oberflächen

Die folgenden Erklärungen zu impliziten Oberflächen und Blending basieren auf Shirley und Marschner [17].

2.3.1 Definition

Eine implizite Oberfläche wird repräsentiert durch Punkte, bei denen die Funktion $f(p)=iso$ ist. Diese Funktion wird auf einen Punkt $p \in R^3$ angewandt, die einen skalaren Wert $iso \in R$ zurückgibt. Diese Funktion lässt sich unterteilen in eine Distanzfunktion $d_i(p)$ und in eine Filterfunktion $g_i(r)$.
$g_i(r)$ ist eine stetig abfallende Funktion mit $g_i(0)=1$ und $g_i(r) \to 0$. Die Variable r ist hierbei der Radius bzw. die Distanz zwischen p und einem Punkt i und entspricht dem Ergebnis der Distanzfunktion $d_i(p)$.
Das Ergebnis einer impliziten Funktion ist eine Kugel mit i als Zentrum (siehe Abbildung 13). Als Analogie kann eine Sonne verwendet werden, die Hitze an die Umgebung abgibt. Im Zentrum ist die Hitze stark und wird schwächer, je weiter die Entfernung zum Zentrum ist.

Abbildung 13: Darstellung des Ergebnisses einer impliziten Funktion in 2D (Bildquelle: nach geisswerks[5]).

Shirley und Marschner [17] nutzen die Notation $f_i(p)=g_i \circ d_i(p)$, in der Regel werden aber beide Funktionen zu einer zusammengefasst. Diese Funktion kann unterschiedliche Formen haben, je nachdem welche Filterfunktion verwendet wird.

2.3.2 Filterfunktionen

5 http://www.geisswerks.com/ryan/BLOBS/keep1a.gif, zuletzt aufgerufen: 08.02.2016

Abbildung 14: Verlauf und Formeln der einzelnen Filterfunktionen (Bildquelle: [17]).

Nach Sherstyuk [30] zufolge, muss eine Filterfunktion eine Gauß-Verteilung besitzen, das heißt die Funktion hat die Form einer Glocke. In Abbildung 14 sind einige Varianten zu sehen. Das Blobby Molecule von Blinn [31] verwendet eine einfache Gaußfunktion, während Metaballs [32] und Soft Objects [33] polynomiale Funktionen nutzen. Obwohl alle vier unterschiedlich aufgebaut sind, unterscheiden sich die Funktionsverläufe kaum voneinander.
Es ist anzumerken, dass alle Funktionen einen zusätzlichen Parameter R besitzen, die in der Definition von impliziten Oberflächen fehlt. Dieser wird *Radius of Influence* genannt [34] und gibt an, bis zu welchem Radius $f_i(p) \neq 0$ ist. Dadurch lässt sich die Größe einer Kugel beeinflussen, was später wichtig zur Nachbildung von Gefäßkrankheiten sein wird.
Sherstyuk [30] testete verschiedene Filterfunktionen auf ihre Geschwindigkeit, wobei der Test sich auf Convolution Surfaces bezog, was eine andere Form von impliziten Oberflächen ist (siehe Abschnitt 2.3.4). Das Ergebnis war, dass Filterfunktionen, die aus polynomialen Funktionen aufgebaut sind, wie Metaballs und Soft Objects, am schnellsten waren.

2.3.3 Blending

Implizite Oberflächen haben die Eigenschaft, dass Flächen, die den gleichen Funktionswert besitzen, miteinander zu einer großen Gesamtfläche verschmelzen, wenn alle Funktionswerte an einem Punkt p aufsummiert werden. Dies wird als *Blending* bezeichnet. Abbildung 15 illustriert diesen Prozess in 2D.

Abbildung 15: Darstellung des Blending Prozesses in 2D. In (a) befinden sich mehrere Punkte, die von ihren Zentren aus jeweils eine implizite Oberfläche definieren. In (b) werden die Grauwerte diskretisiert, wodurch Bereiche mit demselben Grauwert miteinander verschmelzen. In (c) wird alles innerhalb des äußersten Bereichs der Fläche weiß und alles außerhalb als schwarz dargestellt. Dadurch ergeben sich zusammenhängende Flächen (Bildquelle (a-c): geisswerks[6, 7, 8]).

Der große Vorteil von Blending ist, dass es leicht ist, verschiedene Formen von Modellen zu erzeugen, indem dessen Skelett spezifiziert wird. Es müssen dafür lediglich die Positionen der Punktzentren angegeben und die jeweiligen Radius of Influences angepasst werden. Flächen, die ausschließlich aus Punktprimitiven aufgebaut sind, werden als *Potenzialflächen* bezeichnet [35].
Der Nachteil ist allerdings, dass das Skelett auf Punktprimitive beschränkt ist. Potenzialflächen können nicht aus Linien oder Polygonen aufgebaut werden. Möchte der Nutzer also zum Beispiel eine Fläche nachbilden, müssten mehrere Punkte nah beieinander platziert werden, um die Fläche zu approximieren.

2.3.4 Blending bei anderen Skelettprimitiven

Um mehr als nur Punktprimitive nutzen zu können, müssen andere Methoden für das Blending verwendet werden. Bloomenthal und Shoemake [35] beschreiben, welche Möglichkeiten es noch gibt:

Distanzflächen
Hier wird das Maximum aller Funktionswerte an einem Punkt p bestimmt anstatt diese wie bei Potenzialflächen aufzusummieren. Dadurch entsteht die Vereinigung (engl. Union) aller Flächen, die von den Skelettprimitiven generiert werden. Dadurch ist es möglich, Linien oder sogar Polygone zu verwenden, da der Funktionswert nur an dem Punkt des Primitives relevant ist, der am nächsten zu p liegt. Jedoch können bei konkaven Skeletten Falten und Diskontinuitäten entstehen (siehe Abbildung 16). Dies kann umgangen werden, indem die Maxima aller Funktionswerte aufsummiert werden. Das entfernt zwar die Diskontinuitäten, dafür entstehen aber stattdessen Schwellungen an Stellen, wo zwei oder mehr Skelettprimitive aufeinander treffen.

6 http://www.geisswerks.com/ryan/BLOBS/keep1a.gif, zuletzt aufgerufen: 08.02.2016
7 http://www.geisswerks.com/ryan/BLOBS/post6.gif, zuletzt aufgerufen: 08.02.2016
8 http://www.geisswerks.com/ryan/BLOBS/keep1c.gif, zuletzt aufgerufen: 08.02.2016

Abbildung 16: Darstellung der Probleme beim Blending von Distanzflächen. Links ist das Skelett. Rechts ist an der Stelle, wo die beiden Skelettprimitive aufeinanderteffen, eine Falte in der Fläche zu erkennen. Das Aufsummieren der Maxima beider Primitive beseitigt dies, allerdings entsteht dafür an der Stelle eine Schwellung, die in der Mitte zu sehen ist (Bildquelle: [35]).

Convolution Surfaces

Convolution Surfaces sind die Lösung von Bloomenthal und Shoemake [35] für die Probleme von Distanzflächen. Hierbei wird eine Konvolution mit der Filterfunktion ausgeführt. Dies wird ausgedrückt als eine Integration über alle Funktionswerte an einem Punkt p. Das Blending funktioniert hier genauso gut wie bei Potenzialflächen ohne Diskontinuitäten oder Schwellungen und es ist trotzdem möglich, mehrere verschiedene Primitive zu verwenden.
Zudem besitzen Convolution Surfaces die Eigenschaft der Superposition. Das heißt, dass die Konvolution zweier Primitive dieselbe Fläche ergeben, wie die Konvolution von deren Union (siehe Abbildung 17).

Abbildung 17: Darstellung der Superpositons-Eigenschaft von Convolution Surfaces. Das Ergebnis ist gleich, egal ob die Konvolution erst vor oder nach Kombinierung der beiden Skelettprimitive durchgeführt wird (Bildquelle: [35]).

Convolution Surfaces würden sich für die Darstellung von Blutgefäßstrukturen eignen, da Blutgefäße mit Liniensegmenten modelliert werden können. Für diese Arbeit werden allerdings Potenzialflächen verwendet, da diese im Vergleich zu Convolution Surfaces einfacher zu implementieren sind. Zudem reichen Punktprimitive zum Aufbau eines Blutgefäßskelettes dank Blending aus. Darüber hinaus sind die Filterfunktionen laut Sherstyuk [30] schneller bei Punkten als bei Liniensegmenten.

2.3.5 Rendern Impliziter Oberflächen

Es gibt zwei Methoden um implizite Oberflächen zu rendern, die von de Araújo et al. [36] in einer Übersicht zusammengefasst werden.
Eine Methode ist Raytracing. Dies resultiert in präzisen Ergebnissen, allerdings ist es sehr rechenintensiv. Da es für die Arbeit wichtig ist, dass die Anwendung echtzeitfähig ist, wird auf Raytracing verzichtet.
Die andere Methode ist Polygonisierung. Hier wird nach Punkten gesucht, an denen $f(p)=iso$ ist. Diese Punkte werden dann zu Dreiecken verbunden, womit am Ende ein vollständig trianguliertes Mesh entsteht. Polygonisierung lässt sich in drei Vorgehensweisen unterteilen:

Räumliche Dekomposition

Abbildung 18: Lookup-Tabelle der verschiedenen Möglichkeiten, wie Schnittpunkte an Zellen zu Dreiecken beim Marching Cubes Algorithmus verbunden werden können (Bildquelle: [37]).

Hier wird ein Zellengitter in den Raum gelegt, welches das gesamte Objekt umfasst. Es werden anschließend Vertices an den Schnittpunkten der Zellgrenzen mit der Fläche bestimmt und mit Hilfe einer Lookup-Tabelle zu Dreiecken verbunden (siehe Abbildung 18). Der bekannteste Vertreter dieser Vorgehensweise ist der Marching Cubes Algorithmus [37].

Oberflächen-Tracking

Abbildung 19: Darstellung des Oberflächen-Tracking Verfahrens (Quelle: [36]).

Hier wird die Fläche direkt nach relevanten Punkten abgetastet. Es wird an einem Punkt auf der Fläche gestartet und dabei neue Vertices nach bestimmten Einschränkungen generiert. Das daraus entstehende Mesh umwandert damit die Fläche, bis sie es komplett umschlossen hat (siehe

Abbildung 19). Ein Beispiel für Oberflächen-Tracking ist der Marching Triangles Algorithmus [38].

Inflation und Shrinkwrap

Bei *Inflation* [39] werden Teilmodelle innerhalb der impliziten Oberfläche in Relation zu kritischen Punkten generiert. Diese Teilmodelle werden vergrößert, bis sie die Fläche approximieren (siehe Abbildung 20).

Abbildung 20: Darstellung des Inflation-Verfahrens (Quelle: [40]).

Bei *Shrinkwrap* [41] wird die implizite Oberfläche von einer großen Kugel umschlossen. Die Kugel wird nach und nach verkleinert, wodurch sich deren Vertices an die Fläche anschmiegen. Um Details zu approximieren, werden zusätzlich neue Vertices generiert.

3 Konzept

In diesem Kapitel folgt die Beschreibung des Konzeptes der Anwendung, mit der Blutgefäße und Gefäßkrankheiten in 3D skizziert werden sollen. Zuerst wird die verwendete 3D UI beschrieben. Anschließend werden alle Aktionen erklärt, die der Nutzer innerhalb der Anwendung ausführen kann. Zum Schluss wird die Generierung und Darstellung der Blutgefäße mit Metaballs erklärt.

3.1 3D User Interface

Um gezeichnete Blutgefäße darzustellen, wird ein Stereo-Bildschirm mit einer passiven 3D-Brille verwendet. An der Brille befinden sich Infrarot-Marker, die von zwei Infrarot-Kameras am Bildschirm getrackt werden. Dies ermöglicht Bewegungsparallaxe, indem der Nutzer seinen Kopf bewegt. Der Vorteil ist, dass der Nutzer die Gefäße dadurch aus unterschiedlichen Blickwinkeln betrachten kann.
HMDs wären zwar eine Alternative, allerdings haben diese einige Nachteile [19]. Arzt und Patient müssten jeweils ein HMD tragen, wodurch sie sich gegenseitig nicht sehen können. Der PC muss zudem leistungsfähig genug sein, um mit mindestens zwei solcher Geräte arbeiten zu können. Ein weiteres Problem ist, dass das physikalische Eingabegerät nicht sichtbar ist. Es muss also eine virtuelle Repräsentation verwendet werden. Dadurch wird es wichtig, dass die Position des physikalischen und virtuellen Eingabegerätes stets gleich bleibt. Ansonsten kann es zu Fehlern beim Zeichnen kommen. Darüber hinaus haben HMDs einen niedrigen Blickwinkel, was zu Problemen bei der Wahrnehmung von Größe und Distanz führen kann. Aus den Gründen wird deshalb ein Stereo-Bildschirm bevorzugt.
Um Blutgefäße zu zeichnen, wird ein 6-DOF Stylus verwendet, dessen Position und Orientierung von den zwei Infrarot-Kameras am Stereo-Bildschirm getrackt werden. Dies ermöglicht dem Nutzer frei im Raum und aus unterschiedlichen Perspektiven zu zeichnen. Darüber hinaus soll der Nutzer Gefäßsegmente auswählen und manipulieren können. Dies geschieht durch Tasten am Stylus und durch Wechsel von Modi. Auf Gesten wird verzichtet, da der Nutzer diese zuerst lernen muss. Zudem kann es durch Ungenauigkeiten beim Zeichnen zu Schwierigkeiten bei der Interpretation kommen, vor allem in 3D.
Es gibt zwei unterschiedliche Modi. Diese nennen sich „Erstellen“ und „Editieren“. Im Erstellen-Modus kann der Nutzer Linien zeichnen und diese überzeichnen (engl. oversketching), um ihre Form zu verändern. Im Editieren-Modus kann der Nutzer Linien selektieren, löschen und die Blutgefäßdicke verändern.
Die Funktionen können auch anders unterteilt werden. Zum Beispiel kann es je einen Modus für die einzelnen Editier-Operationen geben, in diesem Falle Löschen, Oversketching und Blutgefäßdicke. Zeichnen könnte dann in jedem dieser Modi möglich sein.
Allerdings ist weder die eine noch die andere Unterteilung besser als seine Alternative oder andere Alternativen, da es keine eindeutige Unterteilung gibt. Die erste Unterteilung wurde genommen, damit die Zahl der Modi gering bleibt und um möglichst alle Knöpfe des Stylus zu verwenden.
Die Szene der Anwendung enthält Bedienelemente aus 3D Boxen und 2D Labels, die unten rechts

auf dem Boden platziert sind. Die 3D Boxen agieren analog zu Buttons in 2D. Sie leuchten auf, wenn sie von der Spitze des Stylus berührt werden und sie ändern ihre Farbe, wenn sie aktiviert werden. Die 2D Labels dienen dazu, Informationen über die Anwendung darzustellen. Die Bedienelemente werden zudem von einem transparenten Kasten umschlossen, in der innerhalb nicht gezeichnet werden kann. Dies soll verhindern, dass die Bedienelemente von den Blutgefäßen verdeckt werden können.
In Abbildung 21 sind alle Bedienelemente dargestellt. Ganz oben befindet sich ein Panel mit den beiden Modi, in die gewechselt werden können. Der aktive Modus wird dabei in Gelb angezeigt. Rechts neben dem Panel ist ein weiteres Panel mit Erklärungen zu den Funktionen des Modus. Der Text ändert sich mit dem Wechsel des Modus. Unter dem Panel ist ein Bild des Stylus mit Labeln, die anzeigen, welche Taste welche Funktion besitzt. Die Texte der Labels ändern sich ebenfalls mit dem Wechsel des Modus. Zum Schluss sind ganz unten eine Reset-Box, mit der alle gezeichneten Blutgefäße gelöscht werden können und eine Box rechts daneben, um das Mesh der Blutgefäße in einer hohen Qualität anzuzeigen.

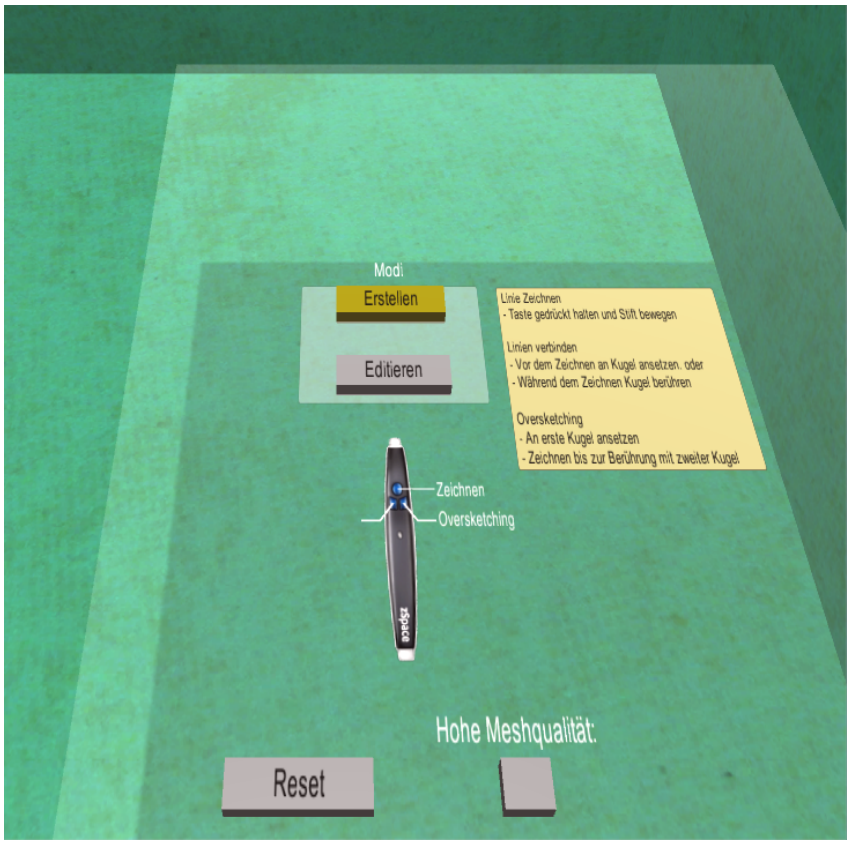


Abbildung 21: Bedienelemente der Anwendung.

Die Vorteile von 3D Boxen sind, dass zum einen die Benutzungsschnittstelle nach konventionellen Richtlinien gestaltet werden kann. Dadurch bekommen Nutzer eine Umgebung, mit der sie bereits vertraut sind. Nachteil ist aber, dass haptisches Feedback bei solchen Boxen fehlt. Es kann vorkommen, dass der Nutzer zu weit durch die Box durchgeht und diese deshalb verfehlt. Dies lässt sich jedoch umgehen, indem die Box groß ist und es anderes Feedback gibt, wenn die Box berührt wird. Ein anderer Nachteil ist, dass gezeichnete Blutgefäße die Sicht auf die Bedienelemente verdecken können. Zwar dient der Kasten dazu, diese Situation zu unterbinden, es kann aber trotzdem zu Verdeckungen kommen, wenn viel in dessen Nähe gezeichnet wird. Eine

Möglichkeit wäre, den Kasten größer zu machen, allerdings würde das zu viel Zeichenraum beanspruchen. Deshalb muss der Nutzer beim Zeichnen entsprechend aufpassen.

3.2 Skizzieren

3.2.1 Verarbeitung der Eingabe

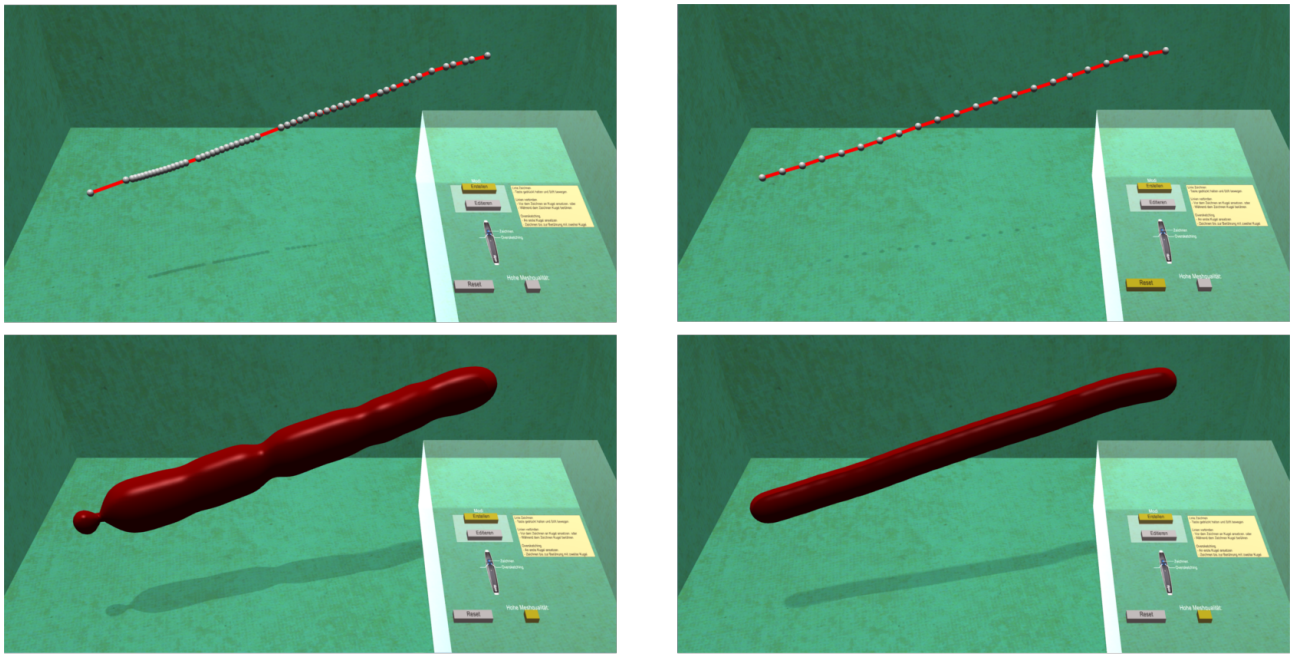
Der Stylus gibt als Eingabe Positionsinformationen in einem 3D Koordinatensystem, die in irregulären Zeitabständen abgetastet werden. Somit ist eine Linie eine zeitabhängige Sequenz von 3D Punkten. Je größer die Abtastrate ist, desto detailreicher ist die Linie.
Beim Zeichnen von Linien kann es viele Ungenauigkeiten geben, die durch Zittern der Hand [10] oder durch Zögern des Nutzers [3] entstehen können. Diese Probleme sind besonders schwerwiegend beim Zeichnen in 3D. Dadurch können schlecht aussehende Gefäße entstehen.
Deshalb wird bei der SBIM-Pipeline (siehe Abschnitt 2.2.1) die Eingabe gefiltert. Von Olsen et al. [10] sind einige Möglichkeiten zur Filterung aufgeführt:

Polyline Approximation
Hier wird nur jeder n-te Punkt als gültige Eingabe genommen. Dies ist zwar eine sehr simple Methode, jedoch basiert die Distribution nur auf globalen Merkmalen.

Resampling
Hier werden alle Punkte verworfen, die unterhalb eines festen Abstandes zum letzten gültigen Punkt liegen. Das heißt, dass die Punkte der Linie immer denselben Abstand haben, egal wie schnell der Nutzer zeichnet.

Curve Fitting
Es werden Bezier- oder B-Splines verwendet, an die sich die Punkte der Linie anfügen. Das führt zwar zu sehr guten Ergebnissen, ist aber rechenaufwendig.

Für die Anwendung wird Resampling verwendet, da es für die Generierung der impliziten Oberfläche wichtig ist, dass die Abstände zwischen den Punkten gleich bleiben (siehe Abbildung 22). Zudem ist diese Methode ein guter Kompromiss zwischen Berechnungsaufwand und Komplexität.



(a) (b)

Abbildung 22: In (a) ist eine Linie ohne und in (b) eine Linie mit Resampling zu sehen. Die abgetasteten Punkte werden durch die weißen Kugeln repräsentiert. Darunter sind die daraus resultierenden Blutgefäße.

Die Filterung führt dazu, dass es nun einfach ist, gerade Linien ohne Krümmungen zu zeichnen. Allerdings können Kurven noch einen zackigen Verlauf haben, weshalb während des Zeichnens eine Gauß-Filterung durchgeführt wird [42]. Hier wird eine neue Position für einen Punkt berechnet, indem dieser mit den vorderen und hinteren Nachbarn aufsummiert wird. Davor wird jede Position mit einem Faktor entsprechend der Gauß-Verteilung multipliziert. Die Gauß-Verteilung wird hierbei diskretisiert, weshalb die Faktoren Binomialkoeffizienten sind.
Die Anzahl der Nachbarn bestimmt dabei die Größe des Filterkerns. Bei der Größe drei werden, während der Nutzer zeichnet, vom letzten Punkt der vordere und hintere Nachbar betrachtet. Bei der Größe fünf werden vom vorletzten Punkt die zwei vorderen und zwei hinteren Nachbarn betrachtet. Abbildung 23 zeigt zwei Linien, auf die ein Gauß-Filter angewendet wurde. Die Kurven der Linie, bei dem die Größe des Filterkerns fünf war, hatten nach der Filterung einen etwas glatteren Verlauf als die Kurven der Linie, bei der die Größe des Filterkerns drei war. Deshalb wird in der Anwendung die Größe fünf bevorzugt.
Abbildung 23 zeigt aber auch, dass die Unterschiede sehr klein sind, wenn der Gauß-Filter nur einmal auf die Linie angewendet wird. Zwar würden durch mehrmaliges Anwenden glattere Kurven entstehen, allerdings würden dabei auch die Abstände zwischen den Punkten schrumpfen. Dies würde zu Problemen beim Verbinden von Linien führen (siehe Abschnitt 4.2.1).

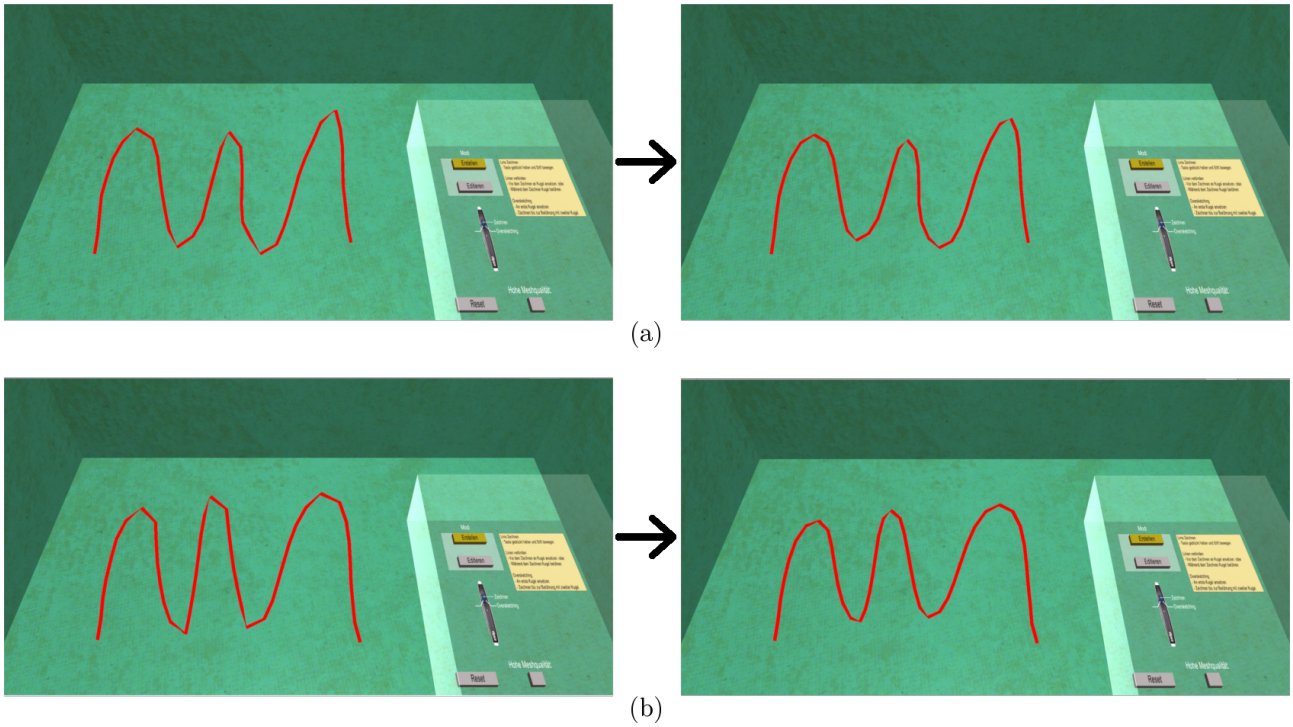


Abbildung 23: Darstellung der Gauß-Filterung von zwei verschiedenen Linien. In (a) wurde ein Filterkern der Größe drei verwendet. In (b) wurde ein Filter der Größe fünf verwendet.

3.2.2 Zeichnen von Linien

Es gibt zwei Möglichkeiten, Blutgefäße zu zeichnen. Entweder es werden zwei Linien gezeichnet, die die Kontur der Blutgefäßwand darstellen, oder es wird nur eine Linie gezeichnet, die das Zentrum bzw. das Skelett des Gefäßes repräsentiert.
Der Vorteil von zwei Linien ist, dass der Nutzer Gefäße unterschiedlicher Breiten zeichnen kann. Zudem wäre die Abbildung von Stenosen und Aneurysmen einfach. Es müsste nur während des Zeichnens die Linien zueinander hin oder weggeführt werden. Jedoch gibt es viele Nachteile. Zum einen können implizite Oberflächen nicht auf den Punkten der Linien platziert werden, da sonst beim Blending zu dicke Blutgefäße entstehen. Stattdessen müsste die Mittellinie bestimmt und die impliziten Oberflächen dort platzieren werden. Das kann aufwendig werden, da die Orientierung der beiden Linien im Raum berücksichtigt werden muss.
Beim Zeichnen von einer Linie gibt es diese Probleme nicht, da die Mittellinie direkt angegeben wird. Zudem hat diese Variante viele Vorteile. Zum einen ist es weniger fehleranfällig, da beim Zeichnen von zwei Linien es zu Kreuzungen und anderen Sonderfällen kommen kann. Zum anderen ist es einfacher, Blutgefäße mit einer gleichmäßigen Dicke zu zeichnen. Bei der anderen Variante ist dies schwerer und es kann dazu führen, dass einige Strukturen als krankhafte Veränderungen fehlinterpretiert werden. Zudem bieten implizite Oberflächen bereits eine gute Möglichkeit, die Blutgefäßdicke zu variieren. Aus diesen Gründen wird deshalb in der Anwendung nur eine Linie gezeichnet.

3.2.3 Verbinden von Linien

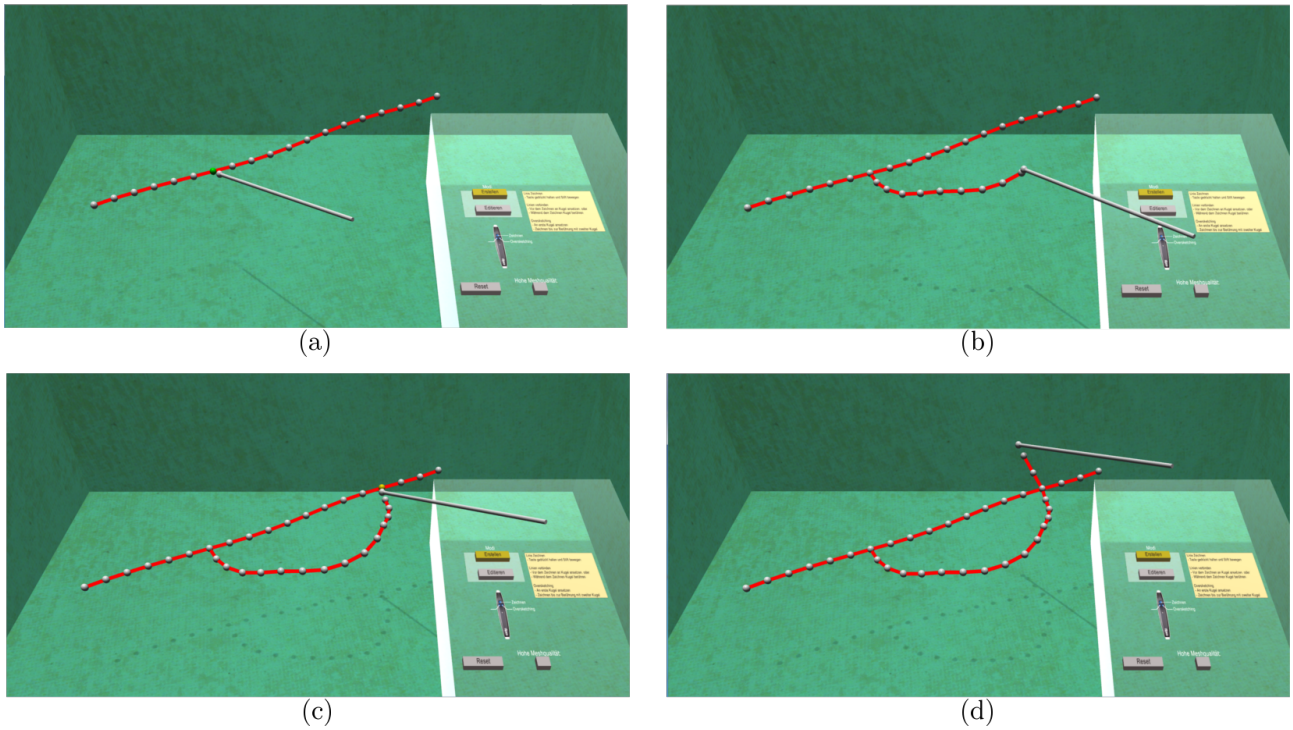


Abbildung 24: Verbinden von Linien. Durch Ansetzen an eine Kugel (a) werden beim Zeichnen die Linien verbunden (b). Durch Berühren weiterer Kugeln beim Zeichnen (c) entstehen weitere Verbindungen (d).

Da sich die Linien im Raum befinden, ist es schwieriger sie beim Zeichnen zu treffen als in 2D. Um das Problem zu lösen, wird an jedem abgetasteten Punkt eine Kugel platziert. Kommt es zur Kollision mit der Spitze des Stylus, dann wird der aktuell abgetastete Punkt auf die Position der kollidierten Kugel gesetzt (siehe Abbildung 24). Dadurch entsteht der Eindruck, dass die Linien dort miteinander verbunden sind. Berührt außerdem der Stylus eine Kugel, ändert sich die Farbe als Feedback. Nachteil ist, dass es keine Verbindungen im Zwischenraum von zwei abgetasteten Punkten geben kann..
Alle Kugeln haben eine feste Größe und lassen sich nicht skalieren. Eine Skalierung würde zur Folge haben, dass der Abstand zwischen den Kugeln angepasst werden muss, da sonst der Stylus mit mehreren gleichzeitig kollidieren würde. Zudem darf der Abstand nicht zu groß sein, sonst kommt es zu Fehlern bei der Erstellung der Blutgefäße (siehe Abschnitt 3.3.1).

3.2.4 Linien Editieren

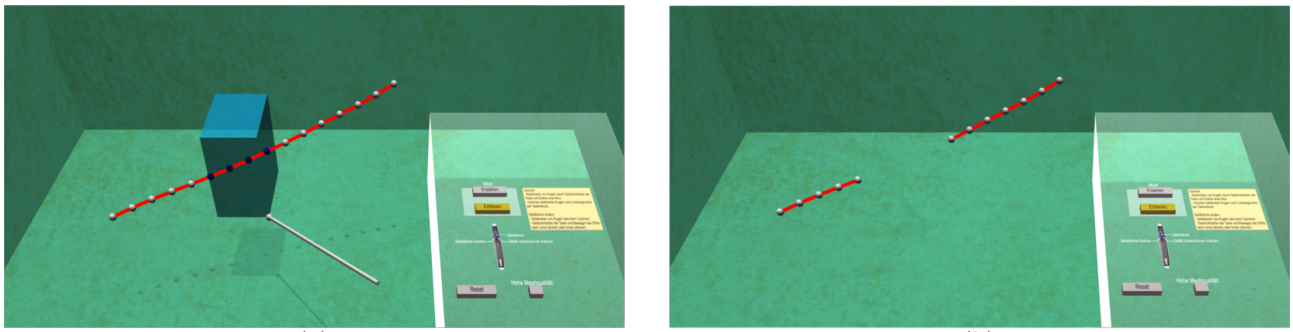
Um Linien zu editieren, stehen dem Nutzer drei Aktionen zur Verfügung:

Selektion
Durch Drücken der entsprechenden Taste wird eine transparente Box aufgezogen, dessen Größe durch Bewegung des Stylus manipuliert werden kann. Alle Kugeln, die sich innerhalb dieser Box befinden, werden markiert und ändern entsprechend ihre Farbe (siehe Abbildung 25 (a)).

Es gibt noch alternativ die Möglichkeit, einen Ellipsoid als Form zu verwenden, mit der die Kugeln entlang der Gefäßrichtung selektiert werden können. Der Vorteil hierbei ist, dass der Nutzer nicht versehentlich Kugeln aus anderen Gefäßsegmenten auswählen kann, so wie es bei einer Box auftreten kann. Ein Ellipsoid hat aber auch zum Nachteil, dass das Auswählen mehrerer separater Gefäßsegmente gleichzeitig nicht möglich ist.
Aus dem Grund wird deshalb die Box bevorzugt. Zudem entspricht sie der Rubberband-Metapher [9] in 3D. Somit wäre Selektieren für die Nutzer natürlicher, da es ähnlich wie das Selektieren in 2D funktioniert.

Löschen

Möchte der Nutzer Liniensegmente löschen, müssen zuerst die Kugeln selektiert werden, die zum Segment gehören. Durch Drücken der entsprechenden Taste werden die Kugeln zusammen mit ihren abgetasteten Punkten entfernt (siehe Abbildung 25 (b)).

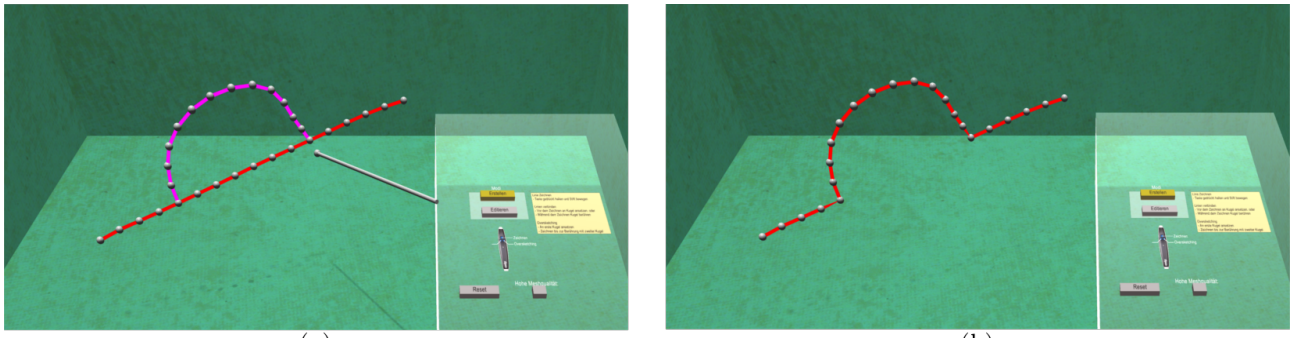


(a) (b)

Abbildung 25: Markieren von Kugeln mit einer aufgezogenenen Box (a) und anschließendes Löschen (b).

Oversketching

Beim Oversketching zeichnet der Nutzer eine Linie, die eine zuvor gezeichnete ersetzen soll. Hierzu wird an eine Kugel der alten Linie angesetzt und per Tastendruck eine neue Linie gezeichnet, bis eine zweite Kugel berührt wird. Danach werden alle abgetasteten Punkte zwischen den berührten Kugeln mit den Punkten der neuen Linie ersetzt (siehe Abbildung 26).
Diese Vorgehensweise hat eine Limitation. Es kann kein Liniensegment überzeichnet werden, wenn es an einem abgetasteten Punkt eine Verbindung mit einer anderen Linie gibt. Verbundene Linien können sich nicht beeinflussen, deshalb würde es beim Oversketching zum Lösen der Verbindung kommen.



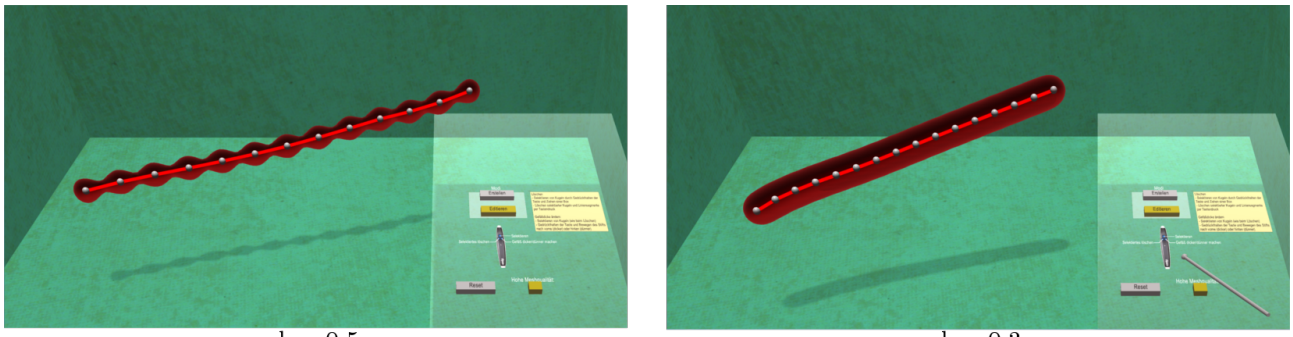
(a) (b)

Abbildung 26: Oversketching einer Linie. Die Oversketching-Linie ist in magenta dargestellt.

3.3 Erzeugung und Darstellung der Blutgefäße

3.3.1 Generierung mit Metaballs

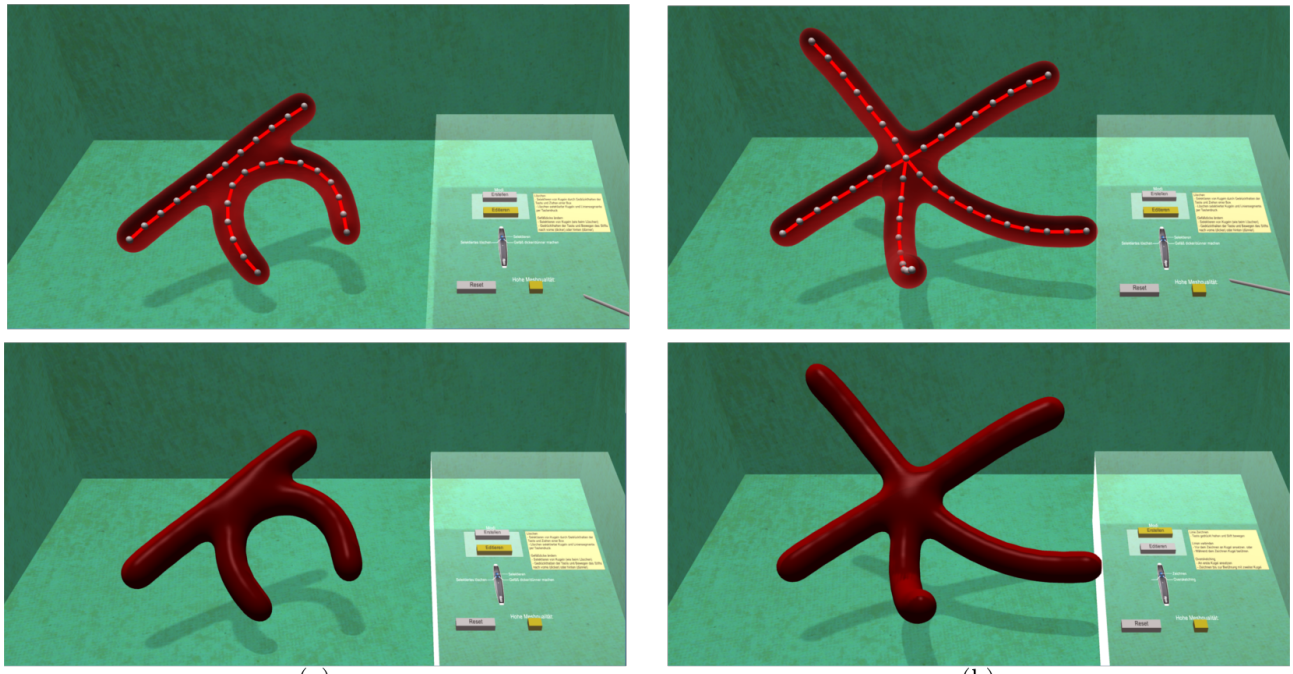
Für die Anwendung werden Metaballs verwendet, dessen Filterfunktion aus Polynomen zweiten Grades aufgebaut ist. Zwar würden Soft Objects, die Polynome sechsten Grades verwenden, glattere Oberflächen generieren, jedoch würden sie bei simplen Strukturen wie Blutgefäße keinen Unterschied machen.
Um ein Blutgefäß zu erzeugen, wird an jedem abgetasteten Punkt ein Metaball platziert. Sind die Metaballs nah genug beieinander, blenden sie zu einer Röhrenstruktur. Hier ist der Abstand zwischen den abgetasteten Punkten wichtig. Ist nämlich der Abstand zu klein, dann gibt es zu viele Metaballs, wodurch die Darstellung berechnungsintensiv wird. Ist der Abstand zu groß, dann entsteht statt einer Röhrenstruktur eine Wellenstruktur (siehe Abbildung 27). Durch Experimentieren ergab sich 0,3 als geeigneter Abstand. Niedrigere Werte gaben keine Verbesserung des Aussehens der Röhrenstruktur. Der Wert 0,4 wäre zwar auch möglich gewesen, jedoch würden Wellentrukturen und Lücken entstehen, sobald die Dicke des Blutgefäßes dünner wird.



d = 0,5 d = 0,3

Abbildung 27: Linien mit unterschiedlichen Abständen d zwischen den abgetasteten Punkten. Links ist der Abstand zu groß, weshalb sich eine Wellenstruktur gebildet hat.

Der große Vorteil ist, dass verzweigte Gefäßstrukturen dank Blending stets glatte Übergänge besitzen. Allerdings kann das Blending auch zu einem Problem werden. Liegt ein Blutgefäß zu nah an einem anderen, dann blenden sie ebenfalls zusammen und das Ergebnis entspricht keinem richtigen Gefäß. Zudem kommt es zu Ausbeulungen an Punkten, an denen mehrere Linien miteinander verbunden sind. Diese Situationen werden als *Unwanted Blending* bezeichnet [43] (siehe Abbildung 28). In der aktuellen Version der Anwendung wird dies nicht unterbunden, deshalb muss vorsichtig gezeichnet werden.



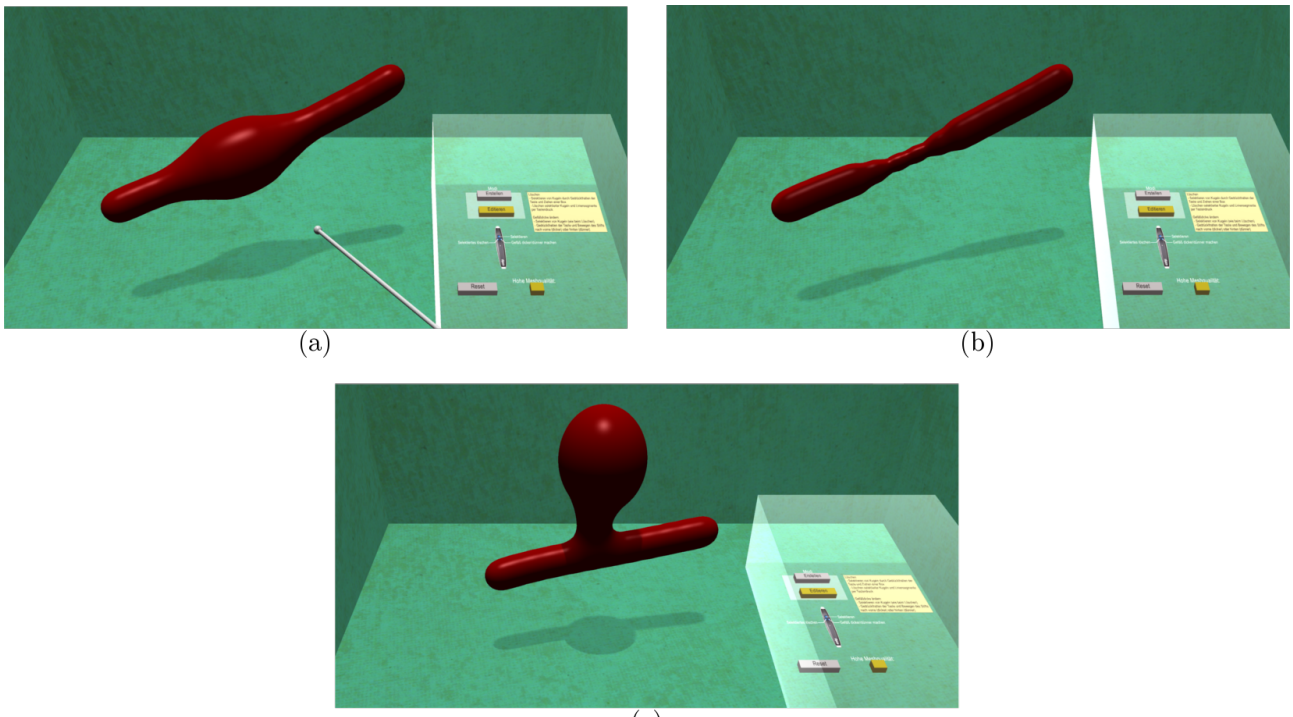
(a) (b)

Abbildung 28: Fälle des Unwanted Blendings. In (a) wurde ein Gefäß zu nah an ein anderes gezeichnet. In (b) entstand eine Schwellung, wo mehrere Linien an einem Punkt verbunden sind.

Für die Polygonisierung wird Marching Cubes verwendet, da räumliche Dekompositionsverfahren nach de Araújo et al. [36] im Vergleich zu anderen Verfahren (siehe Abschnitt 2.3.5) sehr schnell sind. Surface-Tracking Algorithmen generieren Meshes hoher Qualität und Inflation und Shrinkwrap eignen sich zur Rekonstruktion von topologischen Besonderheiten. Sie sind aber dafür berechnungsintensiver. Da es für die Anwendung wichtig ist, dass das Zeichnen interaktiv funktioniert, wird räumliche Dekomposition bevorzugt. Zudem gibt es für Marching Cubes effiziente Implementierungen, da dieser Algorithmus etabliert und weit verbreitet ist.

3.3.2 Erzeugung von Pathologien in Blutgefäßen

Um die Dicke eines Gefäßsegmentes zu verändern, müssen zuerst deren Kugeln selektiert werden. Hält der Nutzer nun die entsprechende Taste gedrückt und bewegt den Stylus nach vorne, dann wird das Gefäß dünner. Bewegt der Nutzer den Stylus nach hinten, dann wird das Gefäß dicker (siehe Abbildung 29). Im Hintergrund wird dabei der Radius of Influence der Metaballs verändert. Als Analogie kann Knete verwendet werden, die zusammengedrückt oder langgezogen wird.



(a) (b) (c)

Abbildung 29: Erzeugung eines spindelförmigen Aneurysmas (a), einer Stenose (b) und eines sackförmigen Aneurysmas (c).

Dank Blending bleibt die Glattheit der Gefäße erhalten. Allerdings muss darauf geachtet werden, dass das Gefäß nicht zu dünn wird. Der Abstand zwischen den abgetasteten Punkten würde sonst zu groß sein und es könnten Wellenstrukturen oder sogar Lücken im Gefäß entstehen. Zudem muss auch hier auf Unwanted Blending geachtet werden. Wird ein Gefäßsegment sehr dick, dann können Verschmelzungen mit umliegenden Gefäßstrukturen entstehen.

3.3.3 Darstellung des Meshes

Das Mesh des Gefäßes wird in einer dunkelroten Farbe dargestellt. Wird das Mesh mit dem Stylus berührt, dann werden die Vorderseiten der Dreiecke, die dem Nutzer zugewandt sind, transparent. Dadurch wird die Mittellinie mit den Kugeln sichtbar. Die sichtbaren Hinterseiten der Dreiecke werden opak mit Shading dargestellt. Zur Transparenz wird ebenfalls gewechselt, wenn gezeichnet oder selektiert wird, um zu verhindern, dass der Nutzer Kugeln verfehlt.

4 Implementierung

In diesem Kapitel wird auf die Implementierung des Konzeptes eingegangen. Es werden zuerst die verwendeten Tools und Frameworks vorgestellt. Anschließend wird beschrieben, wie die im Konzept beschriebenen Aspekte mit ihnen umgesetzt werden.

4.1 Tools und Frameworks

4.1.1 zSpace

Abbildung 30: zSpace 100 Modell mit Stylus und 3D-Brille (Bildquelle: techbriefs[9]).

Das zSpace (siehe Abbildung 30) ist ein Stereo-Bildschirm, dass von zSpace Inc. entwickelt wurde [44]. Es erlaubt Visualisierung und Interaktion mit Objekten in 3D. Zu seinen Komponenten zählen ein stereoskopisches Display, ein 6-DOF Stylus mit drei Tasten und eine passive 3D-Brille mit Infrarot-Markern, um die Position des Kopfes mit Hilfe von zwei Kameras am Bildschirm zu tracken. Dadurch wird ein realitätsnaher Eindruck erzeugt. Es erfüllt somit alle Voraussetzungen um die im dritten Kapitel beschriebene Anwendung umzusetzen.

4.1.2 Unity

Unity[10] ist eine Game Engine zum Entwickeln von Computerspielen in 2D und 3D. Es benutzt C# und Javascript als Skriptsprachen, wobei für die Anwendung C# verwendet wird. Unity bietet eine einfache Integration mit dem zSpace, da von zSpace Inc. ein Plugin dafür bereitgestellt wird.

9 http://www.techbriefs.com/images/stories/NTB/2012/FEATURES/40431-155_fig2.png, zuletzt aufgerufen: 08.02.2016

10 https://unity3d.com/, zuletzt aufgerufen: 08.02.2016

Zudem bietet die Engine viele Features, die das Programmieren vereinfachen. Darüber hinaus gibt es für Unity einen Asset Store. Dies ist ein Online-Laden, in der 3D-Modelle, Texturen und Soundeffekte zur Verwendung in Computerspielen, sowie Skripte und Shader zur Erweiterung der Funktionalitäten der Engine angeboten werden.
Im folgenden sind die zwei wichtigsten Begriffe erläutert, die zum Verständnis der Implementierung erforderlich sind.

Komponente[11]
Komponente sind Funktionalitäten, die einem Objekt zugeteilt werden. Sie können Aussehen, Aktionen und Verhalten definieren. Zu den Komponenten, die am meisten verwendet werden, zählen *Transform*, *Renderer*, *Collider* und *Skripts*.
Die Transform-Komponente definiert die 3D Position, Rotation und Größe eines Objekts. Renderer stellen, je nachdem welcher spezifisch verwendet wird, ein 2D Bild oder ein Mesh dar. Collider definieren einen Bereich, in der Kollisionen stattfinden sollen. Der Programmierer muss dann nur noch angeben, was genau bei einer Kollision passieren soll. Skripts sind der Programmcode, der geschrieben wird. In jedem Skript werden eine oder mehrere Klassen definiert.

GameObject[12]
Ein GameObject ist ein Container für Komponenten. Sie haben standardmäßig immer eine Transform-Komponente. GameObjects können anderen GameObjects als Kinder untergeordnet werden, wodurch das Kind die Eigenschaften des Elternobjektes erbt. Somit lassen sich Hierarchien aufbauen, um Objekte zu ordnen und zu gruppieren.

4.2 Skizzieren

4.2.1 Zeichnen

Jedes einzelne Liniensegment, das der Nutzer zeichnet, ist genau ein GameObject. Es enthält eine *Line Renderer* Komponente, die dafür zuständig ist, eine Linie auf dem Bildschirm darzustellen. Diese Komponente bekommt als Eingabe eine Liste von Vektoren und zeichnet damit eine Linie aus Triangle Strips, die stets der Kamera zugewandt sind. Eine Limitation hierbei ist, dass die Linie keine Lücken haben kann. Deshalb muss für jede neue Linie, die gezeichnet wird, ein neues GameObject erstellt werden.
Weiterhin hat jedes Liniensegment eine Reihe von Kugeln als Kindobjekte, deren Anzahl die Anzahl der Vektoren im Line Renderer entspricht. Die Kugeln werden beim Zeichnen an die Vektorpositionen platziert und agieren als Kommunikationspunkte zwischen dem Stylus und dem Liniensegment. Jede Kugel enthält einen Mesh Renderer, einen Sphere Collider, ein Skript, der Informationen über dessen Zustand enthält, und ein Skript namens MetaballNode, der den Radius of Influence Parameter enthält. Dies ist wichtig für die Generierung der Metaballs später.
Zwischen den Liniensegmenten gibt es keine Hierarchien, wenn Verbindungen existieren. Der Grund dafür ist, das die Unterordnung unklar wird, wenn Gefäßstrukturen, so wie beim *Circle of*

11 http://docs.unity3d.com/Manual/Components.html, zuletzt aufgerufen: 16.02.2015
12 http://docs.unity3d.com/Manual/class-GameObject.html, zuletzt aufgerufen: 16.02.2015

Willis [6], Zyklen enthalten. Zudem müsste die Hierarchie bei manchen Löschungen neu geordnet werden, was bei vielen Linien aufwendig wäre. Deshalb wird auf eine Hierarchie verzichtet. Es bedeutet aber auch, das verbundene Linien sich nicht beeinflussen können. Aus dem Grund kann der Nutzer keine einzelnen Linien verschieben oder rotieren, da es zu unbeabsichtigten Kreuzungen und Lösungen von Verbindungen kommen kann.
Der Stylus ist ein GameObject in der Szene. Es enthält ein Script von zSpace Inc., dass die Position des physikalischen Stylus im Raum bestimmt und stets aktualisiert. Für diese Anwendung wird der Stylus virtuell in die Szene verlängert (siehe Abbildung 31). Dies geschieht durch zwei GameObjects, die Kinder vom Stylus-Objekt sind. Einer ist ein langer Zylinder, der andere ist eine Kugel, die als die Spitze des Stylus fungiert.

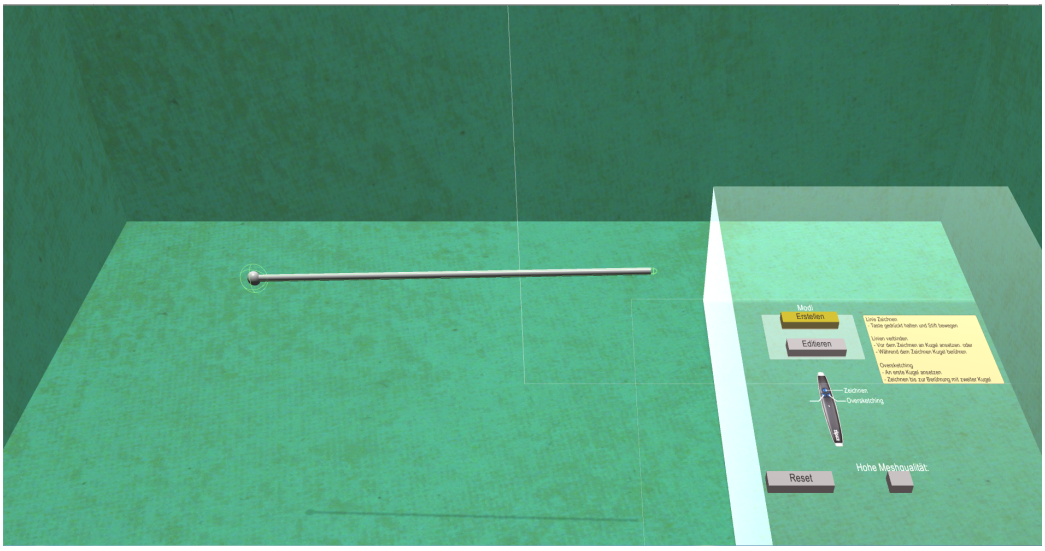


Abbildung 31: Der virtuelle Stylus mit seinen Kollisionsobjekten (gün).

Der Grund für diese Verlängerung soll zum einen sein um zu verhindern, dass der physikalische Stylus und die Hände virtuelle Objekte berühren. Der stereoskopische 3D Effekt ist nur eine Illusion. Möchte der Nutzer z. B. mit der Hand unter ein Gefäß fassen, wird sich die Hand darüber befinden. Somit wäre die Illusion gebrochen und das könnte den Nutzer verwirren.
Zum anderen enthalten der Zylinder und die Kugel jeweils einen Collider. Für die Kugel wurde der Collider so platziert, sodass es zwischen den Kugeln der Linie und dem Stylus zu einer Kollision kommen kann. Dabei ist anzumerken, dass der Collider der Stylusspitze größer ist, damit der Nutzer nicht allzu präzise beim Berühren der Kugeln der Linie sein muss. Die Größe ist so gewählt, sodass die Spitze nicht mit zwei Kugeln gleichzeitig kollidieren kann. Deshalb ist es wichtig, dass die Abstände zwischen den Kugeln bei der Gauß-Filterung nicht zu klein werden. Für den Zylinder dient der Collider, um beim Berühren des Meshes der Gefäße, die Linie und die Kugeln sichtbar zu machen.
Die Spitze der Kugel enthält ein Skript namens *LineDrawer*, der die Funktionen für das Zeichnen von Linien definiert. Auf Tastendruck wird ein Liniensegment-GameObject erstellt. Es wird dann in jedem Frame die Position des Stylus abgefragt. Diese Position wird dem Line Renderer übergeben, wenn dessen Abstand zur letzten gültigen Position einem bestimmten Wert entspricht. Anschließend wird an der Position eine Kugel erstellt. Kommt es während des Zeichnens zu einer Kollision mit einer Kugel einer anderen Linie, dann erfolgt wie in Abschnitt 3.2.3 beschrieben eine Verbindung. In der aktuellen Version der Anwendung ist es nicht möglich, die momentan

gezeichnete Linie mit sich selbst zu verbinden, da die Collider seiner Kugeln erst aktiv werden, wenn mit dem Zeichnen aufgehört wurde.

4.2.2 Editieren

Die Spitze des Stylus enthält neben dem LineDrawer noch ein weiteres Skript namens *LineEditor*, der die Funktionen für das Editieren von Linien definiert.
Beim Selektieren wird per Knopfdruck ein GameObject erstellt, die eine transparente Box enthält. Wenn diese Box mit den Kugeln kollidiert, dann werden sie in eine interne Liste gepackt, die jedes Liniensegment per entsprechendem Skript besitzt.
Wenn nun auf die Taste für das Löschen gedrückt wird, werden die Kugeln innerhalb dieser Listen und die entsprechenden Vektoren des Line-Renderers entfernt. Da Linien im Line Renderer stets kontinuierlich sind, muss beim Entstehen von Lücken eine Sonderbehandlung durchgeführt werden. Zuerst werden alle Kugeln vor und hinter den selektierten Kugeln kopiert. Danach werden zwei neue Liniensegment-GameObjects erstellt und diese dort eingefügt. Anschließend wird das alte Liniensegment-GameObject entfernt.
Das Oversketching funktioniert nach demselben Prinzip wie beim normalen Zeichnen, weshalb dies vom LineDrawer durchgeführt wird. Der Nutzer setzt wie in Abschnitt 3.2.4 beschrieben an eine Kugel an und zeichnet eine neue Linie, bis mit einer zweiten Kugel kollidiert wird. Momentan funktioniert das Oversketching nicht, wenn die beiden Verbindungspunkte von zwei unterschiedlichen Liniensegmenten kommen, da der Algorithmus nur bei einem Liniensegment funktioniert.
Beim Verändern der Blutgefäßdicke werden die Radien aller Kugeln innerhalb der internen Listen angepasst.

4.3 Metaballs Implementierung

Hier wird eine vorhandene Implementierung namens *Skinned Metaballs* von *Junk Games* verwendet[13]. Sie generiert ein zusammenhängendes Mesh aus einer beliebigen Anzahl an impliziten Funktionen per Marching Cubes.
Dies wird definiert durch ein GameObject namens *StaticSeed*, dass ein entsprechendes Skript enthält. Die wichtigen Parameter hier sind die *GridSize* und der *Power Threshold*. Die GridSize beeinflusst die Größe der Zellen für den Marching Cubes Algorithmus. Je kleiner der Wert ist, desto detaillierter ist das Mesh der Blutgefäßwand. Der Power Threshold definiert bis zu welchem Wert iso in $f(p)=iso$ die implizite Oberfläche polygonisiert wird.
Das StaticSeed enthält zudem vier Kindobjekte. Die für die Anwendung relevanten Kinder sind die SourceRoot und das StaticMesh. In die SourceRoot kommen die GameObjects hinein, aus dessen Positionen Metaballs generiert werden sollen. Dies geschieht aber nur, wenn die Kinder ein MetaballNode Skript besitzen. In die SourceRoot kommen alle Liniensegmente hinein. Im StaticMesh wird das Mesh, das durch die Metaballs entsteht, gespeichert.
Das Skript des StaticSeeds hat eine CreateMesh() Funktion, die aus allen Kindern im SourceRoot

13 https://www.assetstore.unity3d.com/en/#!/content/38054, zuletzt aufgerufen: 16.02.2016

das Mesh erstellt. Die Funktion wird aufgerufen, wann immer es an den Liniensegmenten zu irgendwelchen Änderungen kommt und wenn die Dicke einzelner Metaballs manipuliert wird.
Die Implementierung erlaubt es schnell und einfach komplexe Blutgefäßstrukturen zu generieren, allerdings hat sie einige Limitationen. Zum einen entsteht kein zusammenhängendes Mesh, wenn die GridSize zu groß ist (siehe Abbildung 32). Welche GridSize angemessen ist, hängt davon ab, wie dick ein normales Gefäß ist. Dazu kommt noch das Problem, dass der Marching Cubes Algorithmus nach dem Platzieren eines neuen Metaballs das gesamte Mesh neu generiert. Dadurch wird die Generierung berechnungsintensiver, je mehr Metaballs es gibt, vor allem bei einer kleinen GridSize. Irgendwann ist die Interaktivität nicht mehr gewährleistet, da es mitunter ein paar Sekunden dauern kann, bis die Generierung abgeschlossen ist. Deshalb gibt es bei den Bedienelementen die Box für hohe Meshqualität. Der Nutzer zeichnet und manipuliert Gefäße bei einer hohen GridSize und stellt sie beim Aktivieren der Box klein, um das Ergebnis zu betrachten. Diese Limitation bedeutet auch, dass keine dünnen Blutgefäße gezeichnet werden können, da die GridSize entsprechend klein dafür sein muss.

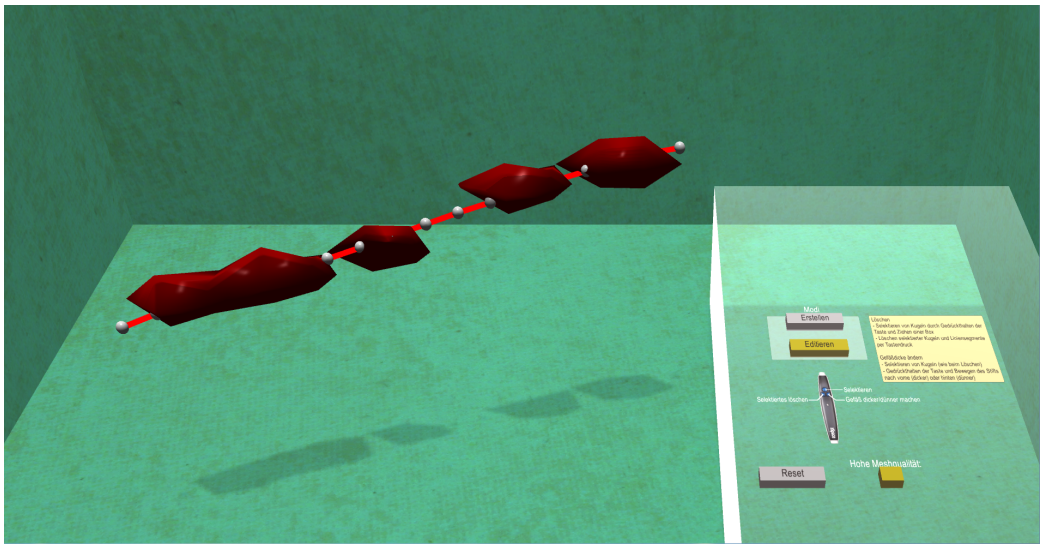


Abbildung 32: Mesh des Blutgefäßes wenn die GridSize zu hoch ist.

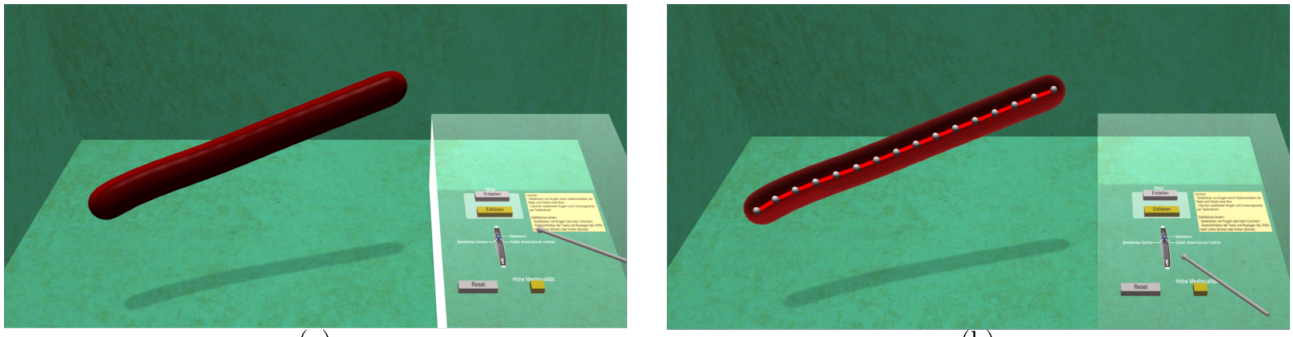
4.4 Shading

Um die Linie und die Kugeln des zugehörigen Gefäßes sichtbar zu machen, wird ein spezieller Fresnel Shader verwendet. Dieser rendert die Vorderseiten der Dreiecke transparent, die senkrecht zur Kamera zeigen. Dadurch sind nur die Vorderseiten am Rand des Meshes und die Hinterseiten der Gefäße sichtbar.
Der Shader wird gesteuert durch drei Parameter:

- Die Farbe der Hinterseiten der Dreiecke,
- die Farbe der sichtbaren Vorderseiten, und
- ein Parameter, der einstellt, wie viele von den Vorderseiten, die nicht senkrecht zur Kamera zeigen, auch transparent gemacht werden sollen.

Im ersten Shader-Durchgang werden die Normalen invertiert, um auf die Rückseiten zuzugreifen. Diese werden der entsprechenden Farbe zugewiesen. Im zweiten Durchgang wird das Skalarprodukt

zwischen der Normale und der Kamerarichtung gebildet Ist das Skalarprodukt eins, dann wird die Vorderseite des Dreiecks transparent dargestellt. Je weiter der Wert gegen null geht, desto opaker werden die Dreiecke.



(a) (b)

Abbildung 33: Darstellung eines Blutgefäßes mit opakem Shader (a) und mit Fresnel-Shader (b).

5 Evaluierung

In diesem Kapitel wird auf die qualitative Evaluierung der Anwendung eingegangen. Es wird zuerst beschrieben, wie der Ablauf ist und welche Methoden verwendet werden. Anschließend werden die Ergebnisse präsentiert und interpretiert. Daraus werden Vorschläge formuliert, wie die Anwendung verbessert werden kann.

5.1 Ablauf

Das Ziel der Evaluierung ist herauszufinden, ob das 3D User Interface der Anwendung dazu geeignet ist, Blutgefäße und Gefäßkrankheiten im Raum zu zeichnen. Dementsprechend stehen hier die User Preference Metrics [19] im Vordergrund. Sie beschäftigen sich damit, wie der Nutzer die Benutzungsschnittstelle wahrnimmt. Das beinhaltetet Aspekte wie Benutzerfreundlichkeit, Lernförderlichkeit, Zufriedenheit, etc. Um diese Metriken zu bestimmen, werden in der Regel Fragebögen verwendet und Interviews durchgeführt. Im folgenden wird der Prozess der Evaluierung genauer beschrieben.
Die Probanden der Evaluierung sind Studierende, von denen keine medizinischen Kenntnisse vorausgesetzt werden. Sie sollen bestimmte Aufgaben mit der Anwendung durchführen. Dem Probanden wird zuerst das Thema der Arbeit erklärt. Anschließend folgt eine Einführung in die Anwendung. Es werden alle Funktionen gezeigt und Szenarien erklärt, die zu Fehlern führen und deshalb vermieden werden sollen. Danach bekommt der Proband ein Aufgabenblatt mit zwei Aufgaben, die in mehrere Unteraufgaben unterteilt sind (siehe Anhang A). Während die Aufgaben durchgeführt werden, soll der Proband seine Gedanken laut äußern und kommentieren. Dies wird als die *Think-Aloud-Methode* bezeichnet [45].
Bei der ersten Aufgabe soll der Proband eine Trifurkation zeichnen, wobei zwei Äste parallel zur Tischplatte und ein Ast in Richtung des Gesichtes gezeichnet werden soll. Nachdem dies getan wurde, soll die Funktion *Hohe Meshqualität* aktiviert und das Ergebnis aus unterschiedlichen Blickwinkeln betrachtet werden. Anschließend soll in den Editieren-Modus gewechselt und einer der Äste gelöscht werden. Mit den verbliebenen Ästen soll dann in einen ein spindelförmiges Aneurysma und in den anderen eine Stenose gezeichnet bzw. erstellt werden.
Bei der zweiten Aufgabe wird alles mit der Reset-Box gelöscht und darauffolgend ein gerades Gefäß gezeichnet. Dieses soll dann mit der Oversketching-Funktion in eine andere Form gebracht werden. Zum Schluss soll an einer beliebigen Stelle ein sackförmiges Aneurysma erstellt werden.
Nachdem alle Aufgaben durchgeführt wurden, füllt der Proband einen Fragebogen aus. Zuerst werden Fragen zum Alter, Geschlecht und Beruf gestellt. Dazu folgen Fragen, ob es Erfahrungen mit Stiftinteraktionen in 3D und mit Grafiktabletts gibt. Zusätzlich wird gefragt, ob der Proband Probleme mit 3D-Sehen hat, weil einige Menschen nicht in der Lage sind, Tiefen angemessen wahrzunehmen. Für sie würde Stereoskopie nicht funktionieren.
Anschließend werden die Fragen zur Anwendung gestellt. Diese sind aus zwei verschiedenen Fragebögen entnommen. Der eine Teil ist der Bogen basierend auf der internationalen Norm ISO 9241/110. Dieser bewertet nach folgenden Kategorien [26]:

- **Selbstbeschreibungsfähigkeit**: Der Nutzer weiß stets, wo er sich in der Software befindet und welche Aktionen ihm zur Verfügung stehen.
- **Aufgabenangemessenheit**: Die Funktionalitäten der Software helfen dem Nutzer seine Aufgaben auszuführen.
- **Steuerbarkeit**: Der Nutzer kann bei der Ausführung seiner Aufgaben den Ablauf einzelner Funktionen beeinflussen.
- **Erwartungskonformität**: Die Aktionen entsprechen bewährten Konventionen und den Erwartungen des Nutzers.
- **Fehlertoleranz**: Der Nutzer kann auf Fehler reagieren und diese mit niedrigem Aufwand beseitigen.
- **Individualisierbarkeit**: Aussehen, Funktionen, etc. der Software lassen sich nach den Präferenzen des Nutzers anpassen.
- **Lernförderlichkeit**: Die Software hilft dem Nutzer, die Funktionen zu lernen und zu beherrschen.

Für den ISO-Norm Fragebogen gibt es eine Langfassung und eine Kurzfassung. Für die Evaluierung wurde die Langfassung genommen, da diese mehr und detailliertere Fragen zu den einzelnen Kategorien enthält.
Es wurden dabei Fragen herausgenommen, die für die Evaluierung der Anwendung irrelevant sind. Zum Beispiel wurden alle Fragen für Individualisierbarkeit entfernt, da der Nutzer keine Anpassungen an der Benutzungsschnittstelle vornehmen kann.
Der zweite Teil des Fragebogens enthält Fragen aus dem Bogen für Präsens in virtuellen Umgebungen von Witmer und Singer [46]. Dieser stellt Fragen zu folgenden Faktoren einer 3D UI:

- **Kontrollfaktoren (CF)**: Diese beschreiben, wie viel Kontrolle der Nutzer über seine Umgebung und Interaktionen hat. Das beinhaltet auch, wie natürlich die Interaktionen sind.
- **Sensorische Faktoren (SF)**: Diese beschreiben, wie viele und wie sehr die Sinne des Menschen in die virtuelle Umgebung einbezogen sind.
- **Ablenkungsfaktoren (DF)**: Diese Faktoren gehen darauf ein, ob der Nutzer in der virtuellen Umgebung immersiert ist und wie sehr die reale Benutzungsschnittstelle wahrgenommen wird.
- **Realismusfaktoren (RF)**: Diese beschreiben, wie real die virtuelle Umgebung für den Nutzer ist und ob sie konsistent zu realen Welt ist.

Der Großteil der Fragen wurde aus dem CF-Bereich entnommen, da diese sich konkret auf die Interaktion mit dem Eingabegerät beziehen. Dazu wurden zwei Fragen aus dem SF-Bereich verwendet, die ermitteln, wie genau und wie gut Objekte aus verschiedenen Blickwinkeln betrachtet werden können. Diese beiden Fragen sollen zur Bewertung der Bewegungsparallaxe dienen. Zusätzlich wurde eine DF-Frage genommen, in der gefragt wird, ob der Eingabemechanismus verwirrend war. Die restlichen Fragen aus den anderen Faktoren beziehen sich auf Aspekte, die eher für HMDs oder andere 3D UIs relevant sind. Deshalb wurden diese entfernt.
Unabhängig von beiden Fragebögen wurde noch hinzugefügt, wie vergleichbar die Bedienelemente zu einer klassischen Benutzungsschnittstelle sind. Diese Frage soll herausfinden, ob 3D-Objekte sich

eignen, um 2D-Objekte wie Buttons nachzubilden.
Beide Fragebögen nutzen eine Sieben-Punkte-Skala, um zu bewerten, wie gut oder schlecht der befragte Aspekt der Software war. Dies erlaubt zudem auch neutrale Aussagen, falls Probanden sich unsicher sein sollten.
Der ISO-Bogen soll allgemeine Fragen zur Usability der Anwendung beantworten, während der Präsens-Bogen Auskunft über die Aspekte geben soll, die sich spezifisch auf 3D UIs beziehen.
Der Fragebogen ist im Anhang der Arbeit zu finden.
Die Fragen werden deskriptiv ausgewertet, dass heißt es werden die Mittelwerte und die Streuung (Standardabweichung) bestimmt.

5.2 Auswertung

Es gab insgesamt neun Probanden, die die Anwendung getestet haben, darunter fünf Männer und vier Frauen im Alter zwischen 18 und 34 Jahren. Dabei waren sieben von ihnen Computervisualistik-Studierende. Die anderen beiden waren jeweils ein Informatik-Student, und ein Doktorand. Drei hatten Erfahrungen mit Stiftinteraktionen in 3D und zwei hatten keine Erfahrungen mit Grafiktabletts. Alle Probanden hatten zudem keine Probleme mit dem 3D-Sehen.

5.2.1 Ergebnisse der Beobachtungen und Think-Aloud Aussagen

3D UI
Alle Probanden fanden den Tiefeneffekt und die Bewegungsparallaxe des zSpace beeindruckend. Allerdings wurde von einigen angemerkt, dass der Tiefeneffekt abnimmt, wenn zu sehr von der Seite geschaut wird. Dies erschwerte das Betrachten der Blutgefäße aus verschiedenen Blickwinkeln. Proband 5 berichtete zudem über leichte Schwindelgefühle, wenn ein Blutgefäß sich nah am Gesicht befand. Darüber hinaus entfernte sich der virtuelle Stylus manchmal vom physikalischen Stylus durch Probleme beim Tracking, was die Probanden verwirrte. Erst durch mehrmaliges Bewegen konnte die korrekte Position wiederhergestellt werden.

Zeichnen
Das Zeichnen der Blutgefäße empfanden alle Probanden als intuitiv. Sie waren alle in der Lage, simple Gefäßstrukturen zu skizzieren, wobei einige dies langsam taten und andere schnell. Jedoch wurde von vier Probanden angemerkt, dass sie das Verbinden der Linien während des Zeichnens als unnatürlich empfanden. Sie erwarteten nämlich, dass die Verbindung sofort beim Berühren der Kugeln erfolgen würde, aber das geschah nur, wenn sich über die Kugel durch bewegt wurde. Dadurch kam es vor, dass die Probanden die Kugeln berührten und dann aufhörten zu zeichnen, wodurch keine Verbindung entstand.
Kam es während des Zeichnens zu Fehlern, entschieden sich ein Großteil der Probanden mit der Reset-Box nochmal von vorne anzufangen, anstatt zu löschen oder die Oversketching-Funktion zu nutzen. Proband 1 erwähnte hierbei, dass eine Funktion zum Abbrechen einer gezeichneten Linie oder zumindest eine Rückgängig-Funktion nützlich wäre. Dies wäre schneller, als das Liniensegment zu selektieren und anschließend zu löschen.

Editieren

Die Editier-Operationen wurden von den Probanden ebenfalls als intuitiv empfunden. Drei Probanden kritisierten allerdings das Fehlen einer Möglichkeit, Gefäße zu rotieren oder zu verschieben. Die Bewegungsparallaxe reichte für sie nicht aus, um Gefäße z. B. von der entgegengesetzten Seite zu betrachten. Zudem kam es manchmal vor, das eine Linie über die Grenzen des Fensters gezeichnet wurde, wodurch sie sich außerhalb der Reichweite des Stylus befand. Proband 2 merkte darüber hinaus an, dass es nützlich wäre, wenn die Richtung eines Liniensegments an einer Kugel umgebogen werden könnte.

Beim Selektieren wurde von Proband 4 erwähnt, dass es bei der Selektionsbox schwierig ist einzuschätzen, wo sich die Hinterseite befindet, da die Kanten nicht hervorgehoben waren. Zudem wurde kritisiert, dass das Mesh nicht transparent blieb, nachdem Kugeln selektiert wurden. Dadurch vergaßen sie schnell, welche Kugeln markiert waren.

Beim Ändern der Blutgefäßdicke merkte Proband 1 an, dass die Veränderung der Dicke zu schnell verlief und er dadurch den Stylus sehr vorsichtig bewegen musste. Zudem empfand er das Zurücksetzen der Blutgefäßdicke beim erneuten Drücken der Taste als irritierend, da für nachträgliche Änderungen von vorne angefangen werden musste.

Viele Probanden hatten zudem Schwierigkeiten mit der letzten Aufgabe, da nicht gesagt wurde, wie sackförmige Aneurysmen zu zeichnen sind. Einige Probanden fanden die richtige Vorgehensweise selbst heraus. Zwei hatten versucht, durch Unwanted Blending beim Verbinden von Linien das Aneurysma nachzubilden, allerdings nur mit mäßigem Erfolg.

Es ist bei zwei Probanden häufig vorgekommen, das die Taste zum Löschen und die Taste für Blutgefäßdicke ändern miteinander vertauscht wurden, da beide Tasten sich dicht nebeneinander befanden. Dadurch wurden versehentlich Liniensegmente gelöscht. Je nachdem wie viel gelöscht wurde, wurde entweder die Linie noch einmal gezeichnet oder komplett von vorne angefangen.

Bedienelemente

Die Probanden konnten mit den Bedienelementen rechts gut umgehen. Einige mussten Funktionen und Vorgehensweisen noch einmal nachschauen. Die Hinweise an den Labels reichten aber aus, um die Funktionen zu verstehen und die Aufgaben ohne fremde Hilfe auszuführen. Lediglich Proband 6 und Proband 7 hatten am Anfang nicht verstanden, das für die Interaktion mit den Bedienelementen die mittlere Taste gedrückt werden musste, da dies nirgends kenntlich gemacht wurde. Proband 2 merkte zudem an, das die Navigation zu den Bedienelementen etwas mühsam war, da er Linkshänder ist.

Es wurde von fast allen Probanden vergessen, die Funktion für hohe Meshqualität wieder auszuschalten, nachdem sie ihr Ergebnis betrachtet hatten. Es wurde erst daran gedacht, nachdem sie mehrfach darauf hingewiesen wurden.

5.2.2 Ergebnisse der Fragebögen

Teil 1 - ISO-Bogen

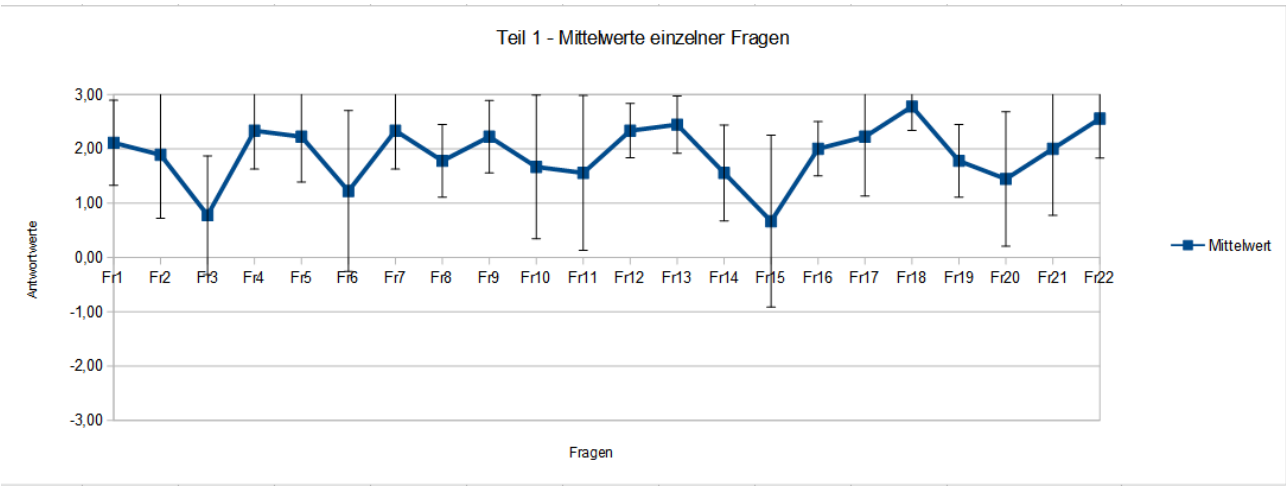


Abbildung 34: Darstellung der Ergebnisse der Fragen des ersten Teils, die über alle neun Probanden gemittelt wurden. Die Streuung wird von den vertikalen schwarzen Linien an jedem blauen Punkt repräsentiert.

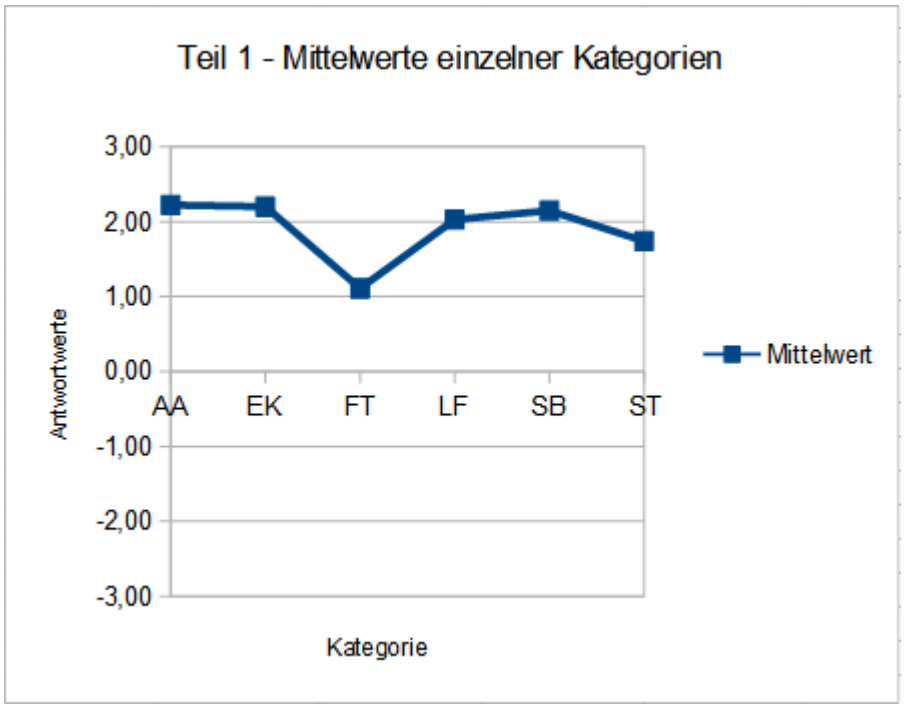


Abbildung 35: Darstellung der Ergebnisse der Kategorien Aufgabenangemessenheit (AA), Erwartungskonformität (EK), Fehlertoleranz (FT), Lernförderlichkeit (LF), Selbstbeschreibungsfähigkeit (SB) und Steuerbarkeit (ST). Die Werte wurden über die Fragen aus den einzelnen Kategorien über alle neun Probanden gemittelt.

Der Großteil der Fragen wurde positiv beantwortet (siehe Abbildung 34). Frage 3 und Frage 15 aus der Kategorie Fehlertoleranz hatten hierbei den niedrigsten Mittelwert. Bei Frage 3 sollte herausgefunden werden, ob Fehler einen hohen Korrekturaufwand erfordern. Frage 15 sollte ermitteln, ob kleine Fehler schwerwiegende Folgen haben. Beide Fragen hatten zudem eine hohe Streuung. Probanden, die keine Fehler machten, bewerteten positiv, während die Probanden, die Fehler machten, negativ bewerteten. Daraus lässt sich schließen, dass Fehler zwar vermeidbar sind, aber dafür schwer zu beseitigen sind, wenn sie auftreten.

Frage 18, die herausfinden sollte, ob die Software sich nach einem einheitlichen Prinzip bedienen lässt, hatte den höchsten Mittelwert. Daraus folgt, dass die Bedienung der einzelnen Funktionen sich nicht zu sehr voneinander unterschieden und dass sie nach denselben Interaktionsmustern ausgeführt werden konnten.
Fragen 6, 10 und 11 hatten eine hohe Streuung. Frage 6 beschäftigte sich damit, ob leicht zwischen Menüs und Masken gewechselt werden kann. Bei Frage 10 wurde thematisiert, ob die Software in zureichendem Maße darüber informiert, welche Aktionen gerade ausgeführt werden. Frage 11 versuchte herauszufinden, ob über fehlerhafte Eingaben zu spät informiert wird. Hier waren sich die Probanden uneinig, ob das Wechseln der Modi mühsam war und ob das Feedback der Bedienelemente sich gut genug hervorhob.
Bei den Ergebnissen für die einzelnen Kategorien (siehe Abbildung 35), hatte Fehlertoleranz den niedrigsten Mittelwert, was auf die Folgen von Fehlern zurückzuführen ist. Aufgabenangemessenheit hatte den größten Mittelwert. Daraus lässt sich schließen, dass mit der 3D UI die Aufgaben gut gelöst werden konnten.

Teil 2 - Präsens-Bogen

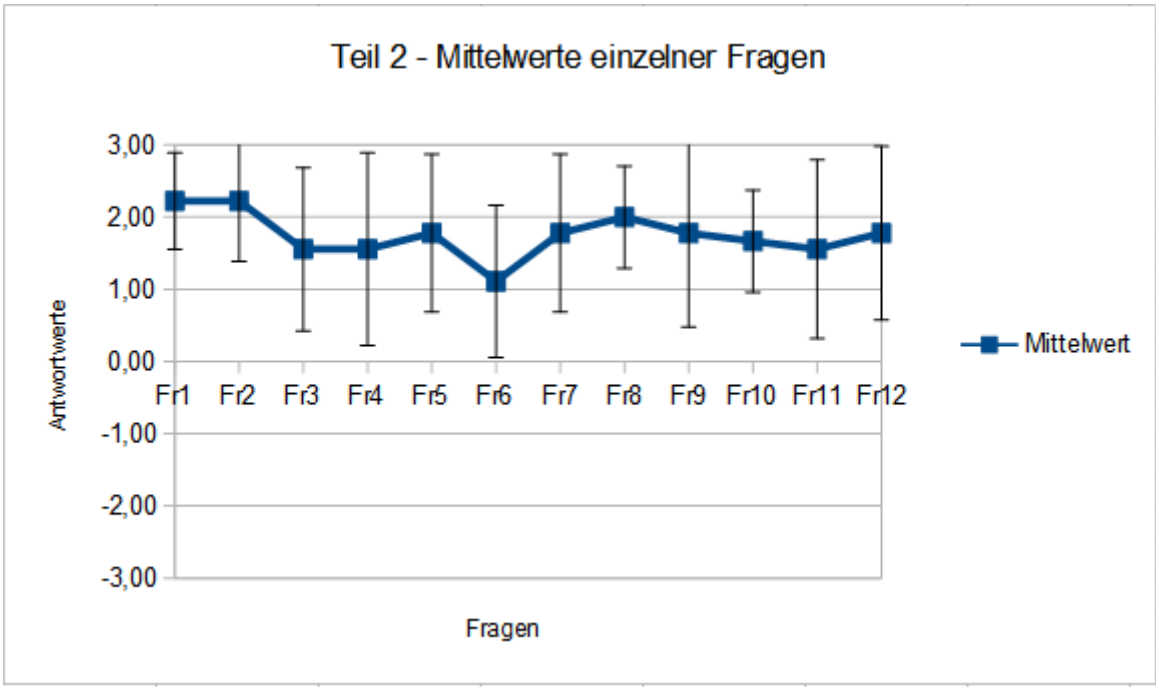


Abbildung 36: Darstellung der Ergebnisse der Fragen des zweiten Teils, die über alle neun Probanden gemittelt wurden. Die Streuung wird von den vertikalen schwarzen Linien an jedem blauen Punkt repräsentiert.

Auch beim Präsens-Fragebogen wurden die Fragen positiv beantwortet (siehe Abbildung 36). Daraus lässt sich schließen, dass die Probanden mit den Eingabegeräten und Ausgabegeräten die Blutgefäße gut zeichnen und manipulieren konnten und dass die Interaktionstechniken als natürlich empfunden wurden.
Frage 6 hatte hierbei den niedrigsten Mittelwert. Hier sollte ermittelt werden, wie gut das Blutgefäß aus mehreren Blickwinkeln betrachtet werden konnte. Dies lässt sich auf die Probleme des Tiefeneffekts beim seitlichen Betrachten zurückführen.
Frage 12, bei der gefragt wurde, ob die Bedienelemente rechts vergleichbar zu einer klassischen

Benutzungsschnittstelle sind, wurde positiv beantwortet, wodurch sich 3D-Objekte gut eignen, um Buttons und andere 2D-Objekte auf 3D-Interaktionen zu übertragen.

5.2.3 Interpretation

Aus den Ergebnissen der Fragebögen lässt sich schlussfolgern, dass das zSpace mit den verwendeten Interaktionstechniken sich gut eignet, um Blutgefäße und Gefäßkrankheiten im Raum zu zeichnen. Da zudem die Probanden die Bedienelemente rechts vergleichbar zu einer klassischen Benutzungsschnittstelle fanden, können deren Richtlinien beim Design genutzt werden, um die Schnittstellen bei 3D UIs zu gestalten.
Allerdings ist die Anwendung anfällig auf Fehler und es fehlen genügend Möglichkeiten diese zu korrigieren. Die Möglichkeiten, die vorhanden sind, werden oftmals nicht benutzt, wodurch sich darauf schließen lässt, dass diese zu mühsam für die Beseitigung der Fehler sind.
Durch die Probleme des Tiefeneffekts beim seitlichen Betrachten wurde festgestellt, dass das zSpace nur von einer Person effektiv genutzt werden kann. Zum einen funktioniert die Bewegungsparallaxe nur mit einer Person. Arzt und Patient müssten also ständig die Brillen und ihre Positionen tauschen. Zum anderen kann der Patient durch diese Probleme dem Arzt beim Zeichnen nur schwer von der Seite zuschauen.

5.2.4 Verbesserung der Anwendung

Zeichnen
Um das Skizzieren von Blutgefäßen zu verbessern, kann als Funktion eingebaut werden, dass die Teillinie verschwindet, wenn beim Zeichnen sich in die entgegengesetzte Richtung bewegt wird. Dadurch können entstandene Fehler schnell behoben werden. Somit müsste die Linie nicht mehr komplett gelöscht und von vorne angefangen werden.
Weiterhin soll beim Verbinden von Linien die Verbindung sofort beim Berühren der Kugeln erfolgen, wodurch die Aktion näher den Erwartungen der Probanden entsprechen würde. Es kann außerdem nicht mehr vorkommen, dass beim Aufhören des Zeichnens die Verbindung nicht erfolgt.
Zusätzlich soll es unmöglich sein, bei hoher Meshqualität zu zeichnen, damit die Nutzer nicht vergessen es auszustellen, nachdem sie ihr Ergebnis betrachtet haben. Es soll zudem ein entsprechender Hinweis dafür eingeblendet werden. Alternativ könnte die Funktion auch automatisch deaktiviert werden, sobald entweder gezeichnet oder editiert wird.

Editieren
Am wichtigsten wäre eine Funktion, um Blutgefäße zu verschieben und zu rotieren. Da sich verbundene Linien nicht beeinflussen können, wird jeweils nur das Skelettobjekt rotiert oder verschoben. Dadurch kann der Nutzer auf Gefäßsegmente zugreifen, die sich außerhalb der Reichweite des Stylus befinden. Zudem können dadurch auch Gefäße aus Blickwinkeln betrachtet werden, die mit der Bewegungsparallaxe schwer erreichbar sind.
Weiterhin soll es möglich sein, den Verlauf einer Linie an einer Kugel zu verändern, um mehr Möglichkeiten zu geben, Linien anzupassen. Für die Selektionsbox soll zudem das

Drahtgittermodell (engl. Wireframe) angezeigt werden, damit die Seiten in der Tiefe besser sichtbar sind.
Zusätzlich soll der Skalierungsfaktor für die Veränderung der Dicke der Blutgefäße kleiner gemacht werden, damit Veränderungen nicht zu schnell erfolgen. Darüber hinaus soll die Dicke nicht wieder zurückgesetzt werden, wenn die Taste ein weiteres Mal gedrückt wird.

Bedienelemente
Zur Verbesserung könnten Bedienelemente eingebaut werden, mit der der Nutzer Aktionen rückgängig machen und wiederherstellen kann. Dadurch lassen sich Fehler besser korrigieren. Zusätzlich soll es eine Option geben, um alle Bedienelemente auf die linke Seite zu platzieren. Somit wären sie für Linkshänder besser erreichbar. Weiterhin soll kenntlich gemacht werden, welche Taste gedrückt werden muss, um mit den Bedienelementen zu interagieren. Die Labels am Bild des Stylus können sich entsprechend verändern, wenn der Kasten um die Bedienelemente berührt wird.

6 Zusammenfassung und Ausblick

6.1 Zusammenfassung

In dieser Arbeit wurde mit dem zSpace und mit der Game Engine Unity eine Anwendung entwickelt, mit der der Nutzer Blutgefäße und Gefäßkrankheiten im Raum zeichnen und manipulieren kann.
Zur Generierung wurden implizite Oberflächen verwendet, die durch Marching Cubes polygonisiert wurden. Durch die Blending-Eigenschaft können Röhrenstrukturen durch ein Skelett aus Punktprimitiven gebildet werden. Dies macht jedoch die Form des Meshes abhängig vom Abstand zwischen den abgetasteten Punkten der gezeichneten Linie. Dadurch muss stets darauf geachtet werden, das bei der Erstellung von Gefäßverengungen keine Wellenstrukturen entstehen. Ein weiteres Problem ist Unwanted Blending, die zu Verschmelzungen naher Gefäßstrukturen und zu Ausbeulungen an Verzweigungen führen. Darüber hinaus hat der Maching Cubes Algorithmus Schwierigkeiten bei der Generierung dünner Gefäße, da diese eine kleine Größe der Zellen brauchen und somit sehr rechenintensiv sind. Dadurch funktioniert das Zeichnen nicht mehr interaktiv.
Die qualitative Evaluierung mit der Think-Aloud Methode und mit Fragebögen hat gezeigt, dass das zSpace und die verwendeten Interaktionstechniken sich gut eignen, um simple Gefäßstrukturen und Krankheiten, wie Aneurysmen und Stenosen, zu skizzieren. Allerdings ist die Anwendung anfällig auf Fehler, die durch Verzeichnen, Unwanted Blending und durch Unachtsamkeit beim Editieren verursacht werden können. Es fehlen dabei genügend Möglichkeiten, solche Fehler zu korrigieren. Dadurch kann das Zeichnen und Editieren umständlich werden, wenn solche Fehler oft auftreten.
Die Evaluierung hat zudem auch gezeigt, dass das zSpace nur von einer Person gleichzeitig effektiv genutzt werden kann. Zum einen nimmt nämlich der Tiefeneffekt beim seitlichen Betrachten ab und zum anderen können die Kameras am zSpace nur eine Brille gleichzeitig tracken. Dadurch funktioniert die Bewegungsparallaxe nur bei einer Person. Somit wäre das zSpace nicht optimal geeignet, um die Aufklärung des Patienten durch den Arzt zu unterstützen.

6.2 Ausblick

Convolution Surfaces könnten als Alternative zu Potenzialflächen genutzt werden, da sie neben Punkten auch Linien als Primitive verwenden können. Dadurch ließe sich ein Segment beliebiger Länge selektieren und verändern, ohne das dazwischen Wellenstrukturen entstehen.
Von Steffen Oeltze [47] wurde zudem ein Ansatz beschrieben, wie das Unwanted Blending Problem bei Convolution Surfaces unterbunden werden kann. Die Filterfunktion wird so modifiziert, sodass der Funktionsverlauf einen steileren und schnelleren Abstieg besitzt. Dadurch tritt Unwanted Blending zum einen erst bei kürzeren Distanzen zwischen Gefäßen auf. Zum anderen werden Schwellungen an Stellen vermieden, wo mehrere Gefäße miteinander verbunden sind. Weitere Möglichkeiten zur Unterbindung von Unwanted Blending sind von Opalach und Maddock [43] und

von Cani und Hornus [48] beschrieben.
Um dünne Gefäßstrukturen darzustellen, könnten räumliche Dekompositionsverfahren verwendet werden, die Oct-Trees nutzen [36]. Dadurch wäre das Mesh nur an den dünnen Gefäßen detailliert. Alternativ können auch Verfahren aus anderen Bereichen, wie Surface-Tracking verwendet werden, so lange sie das Mesh schnell generieren. De Araújo et al. [36] gehen hierbei auf aktuelle Algorithmen zur Polygonisierung von impliziten Oberflächen ein und kategorisieren sie nach:

- wie schnell das Mesh generiert wird,
- wie präzise das Mesh generiert wird,
- wie gut die Qualität des Meshes ist und
- wie sehr topologische Besonderheiten berücksichtigt werden.

Es muss aber dabei darauf geachtet werden, dass das Mesh nur an den Stellen neu generiert wird, an denen es zu Änderungen an der Gefäßstruktur kam.
Die Blutgefäße in der Anwendung werden lediglich in einer Farbe dargestellt, was visuell wenig ansprechend ist. Es könnten stattdessen Methoden verwendet werden, die eine skizzenhafte Darstellung mit einem Bleistift simulieren [49]. Alternativ kann per Texturen, Normal Maps und Bump Maps das Gefäß möglichst real repräsentiert werden.
Das größte Problem am aktuellen Aufbau des 3D UIs ist, dass nur eine Person gleichzeitig die Stereoskopie und die Bewegungsparallaxe nutzen kann. Zukünftige Arbeiten könnten andere Ausgabegeräte wie Head Mounted Displays verwenden und vergleichen, ob diese besser geeignet wären. Alternativ könnte die aktuelle Umgebung so angepasst werden, dass zwei Betrachter gleichzeitig die Eigenschaften von Stereo-Bildschirmen nutzen können. Zum Beispiel könnten aktive Brillen verwendet werden. Hierbei würden sich die Verschlussklappen an den Brillen der beiden Betrachter abwechselnd schließen, wodurch der eine nur das linke Bild und der andere nur das rechte Bild sieht. Dies wäre aber auch mit passiven Brillen möglich, indem zwei Brillen jeweils nur eine Polarisation haben.
Falls die aktuelle 3D UI bestehen bleibt, können zukünftige Arbeiten als nächstes Behandlungsmethoden und Blutflusssimulation in 3D umsetzen. Es müssen lediglich weitere Modi hinzufügt und die entsprechenden Funktionen den Stylustasten zugeordnet werden. Behandlungsmethoden für Aneurysmen und Stenosen, zu denen Stents, Coils und Clips zählen [50], können dann jeweils ein GameObject sein, die mit Hilfe von Unitys Kollisionsmodell mit der Blutgefäßwand interagieren. Zudem kann versucht werden, das Platzieren der Objekte möglichst ähnlich zur echten Operation umzusetzen, um Patienten den Verlauf der Behandlung zu illustrieren. Blutfluss könnte zudem durch Unitys Partikelsystem simuliert werden, die für Kollisionen mit anderen Objekten angepasst werden können.

Literaturverzeichnis

[1] S. Mendis, P. Puska, B. Norrving, World Health Organization, World Heart Federation, und World Stroke Organization, Hrsg., *Global atlas on cardiovascular disease prevention and control*. Geneva: World Health Organization in collaboration with the World Heart Federation and the World Stroke Organization, 2011.

[2] B. Keulers, „Computer-based patient education: its potential in general and plastic surgery“, PhD Thesis, University Nijmegen, 2008.

[3] D. Avola, M. C. Caschera, F. Ferri, und P. Grifoni, „Ambiguities in Sketch-Based Interfaces“, in *2014 47th Hawaii International Conference on System Sciences*, Los Alamitos, CA, USA, 2007, Bd. 0, S. 290b.

[4] P. Saalfeld, A. Baer, U. Preim, B. Preim, und K. Lawonn, „Sketching 2D Vessels and Vascular Diseases with Integrated Blood Flow“, in *Proceedings of the 10th International Conference on Computer Graphics Theory and Applications (GRAPP)*, Berlin, 2015, S. 379–390.

[5] P. Saalfeld, „Methoden zur Skizzierung von Gefäßen und Gefäßerkrankungen mit integriertem Blutfluss“, Masterarbeit, Otto-von-Guericke Universität, Magdeburg, 2014.

[6] K. Zilles, B. N. Tillmann, und Zilles-Tillmann, *Anatomie: mit 121 Tabellen*. Heidelberg: Springer, 2010.

[7] U.-N. Riede, M. Werner, N. Freudenberg, J. P. Baak, und Riede-Werner-Freudenberg, Hrsg., *Basiswissen allgemeine und spezielle Pathologie: mit 76 Tabellen und 47 authentischen Fällen; [mit Fallquiz]*. Heidelberg: Springer Medizin, 2009.

[8] T. Standl und C. Lussi, Hrsg., *Ambulantes Operieren: Rahmenbedingungen - Organisation - Patientenversorgung; mit 23 Tabellen*, 2. Aufl. Berlin: Springer Medizin, 2012.

[9] B. Preim und R. Dachselt, *Interaktive Systeme*, 2. Aufl., Bd. 1. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010.

[10] L. Olsen, F. Samavati, M. C. Sousa, und J. Jorge, „A taxonomy of modeling techniques using sketch-based interfaces“, in *Eurographics State of the Art Report*, 2008.

[11] J. Jorge und F. Samavati, *Sketch-based Interfaces and Modeling*, 1st Aufl. Springer Publishing Company, Incorporated, 2010.

[12] X. Xu, W. Liu, X. Jin, und Z. Sun, „Sketch-based user interface for creative tasks“, in *Proceedings of the 5th Asia Pacific conference on computer human interaction, Beijing*, 2002, S. 560–570.

[13] T. M. Sezgin und A. Davis, „R.: Scale-space based feature point detection for digital ink“, in *In Making Pen-Based Interaction Intelligent and Natural, AAAI Spring Symposium*, 2004.

[14] A. Alexe, V. Gaildrat, und L. Barthe, „Interactive Modelling from Sketches Using Spherical

Implicit Functions“, in *Proceedings of the 3rd International Conference on Computer Graphics, Virtual Reality, Visualisation and Interaction in Africa*, New York, NY, USA, 2004, S. 25–34.

[15] T. Igarashi, S. Matsuoka, und H. Tanaka, „Teddy: A Sketching Interface for 3D Freeform Design“, in *ACM SIGGRAPH 2007 Courses*, New York, NY, USA, 2007.

[16] J. J. Cherlin, F. Samavati, M. C. Sousa, und J. A. Jorge, „Sketch-based Modeling with Few Strokes“, in *Proceedings of the 21st Spring Conference on Computer Graphics*, New York, NY, USA, 2005, S. 137–145.

[17] P. Shirley und S. Marschner, *Fundamentals of Computer Graphics*, 3rd Aufl. Natick, MA, USA: A. K. Peters, Ltd., 2009.

[18] O. Karpenko, J. F. Hughes, und R. Raskar, „Free-form sketching with variational implicit surfaces“, in *Computer Graphics Forum*, 2002, Bd. 21, S. 585–594.

[19] D. A. Bowman, E. Kruijff, J. J. LaViola, und I. Poupyrev, *3D User Interfaces: Theory and Practice*. Redwood City, CA, USA: Addison Wesley Longman Publishing Co., Inc., 2004.

[20] D. A. Bowman, E. Kruijff, J. J. LaViola, und I. Poupyrev, „An Introduction to 3-D User Interface Design“, *Presence Teleoper Virtual Env.*, Bd. 10, Nr. 1, S. 96–108, Feb. 2001.

[21] G. Welch und E. Foxlin, „Motion Tracking: No Silver Bullet, but a Respectable Arsenal“, *IEEE Comput Graph Appl*, Bd. 22, Nr. 6, S. 24–38, Nov. 2002.

[22] H. Perkunder, J. H. Israel, und M. Alexa, „Shape Modeling with Sketched Feature Lines in Immersive 3D Environments“, in *Proceedings of the Seventh Sketch-Based Interfaces and Modeling Symposium*, Aire-la-Ville, Switzerland, Switzerland, 2010, S. 127–134.

[23] A. Nealen, T. Igarashi, O. Sorkine, und M. Alexa, „FiberMesh: Designing Freeform Surfaces with 3D Curves“, in *ACM SIGGRAPH 2007 Papers*, New York, NY, USA, 2007.

[24] B. Laundry, M. Masoodian, und B. Rogers, „Interaction with 3D Models on Large Displays Using 3D Input Techniques“, in *Proceedings of the 11th International Conference of the NZ Chapter of the ACM Special Interest Group on Human-Computer Interaction*, New York, NY, USA, 2010, S. 49–56.

[25] Y. Chen, J. Liu, und X. Tang, „Sketching in the air: a vision-based system for 3D object design“, in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, 2008, S. 1–6.

[26] A. M. Heinecke, *Mensch-Computer-Interaktion*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012.

[27] P. G. Poison, P. G. Polson, P. G. Polson, C. Lewis, C. Lewis, C. Lewis, J. Rieman, J. Rieman, J. Rieman, C. Wharton, C. Wharton, und C. Wharton, *Cognitive Walkthroughs: A Method for Theory-Based Evaluation of User Interfaces*. 1991.

[28] J. Nielsen und R. Molich, „Heuristic Evaluation of User Interfaces“, in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, New York, NY, USA, 1990, S.

249–256.

[29] B. Preim und R. Dachselt, *Interaktive Systeme*, 2. Aufl., Bd. 2. Berlin, Heidelberg: Springer Berlin Heidelberg, 2015.

[30] A. Sherstyuk, „Kernel functions in convolution surfaces: a comparative analysis“, *Vis. Comput.*, Bd. 15, Nr. 4, S. 171–182, Juli 1999.

[31] J. F. Blinn, „A generalization of algebraic surface drawing“, *ACM Trans. Graph. TOG*, Bd. 1, Nr. 3, S. 235–256, 1982.

[32] H. Nishimura, M. Hirai, T. Kawai, T. Kawata, I. Shirkawa, und K. Omura, „Object modeling by distributed function and a method of image generation“, *ResearchGate*, Bd. 68, Nr. 4, Jan. 1985.

[33] G. Wyvill, C. McPheeters, und B. Wyvill, „Data structure for soft objects“, *Vis. Comput.*, Bd. 2, Nr. 4, S. 227–234, Aug. 1986.

[34] J. Bloomenthal und B. Wyvill, Hrsg., *Introduction to Implicit Surfaces*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1997.

[35] J. Bloomenthal und K. Shoemake, „Convolution Surfaces“, in *Proceedings of the 18th Annual Conference on Computer Graphics and Interactive Techniques*, New York, NY, USA, 1991, S. 251–256.

[36] B. R. de Araújo, D. S. Lopes, P. Jepp, J. A. Jorge, und B. Wyvill, „A Survey on Implicit Surface Polygonization“, *ACM Comput Surv*, Bd. 47, Nr. 4, S. 60:1–60:39, Mai 2015.

[37] W. E. Lorensen und H. E. Cline, „Marching Cubes: A High Resolution 3D Surface Construction Algorithm“, in *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques*, New York, NY, USA, 1987, S. 163–169.

[38] A. Hilton, J. Illingworth, und others, „Marching triangles: Delaunay implicit surface triangulation“, *Univ. Surrey*, 1997.

[39] B. T. Stander und J. C. Hart, „Interactive re-polygonization of blobby implicit curves“, in *Proceedings of the Western Computer Graphics Symposium*, 1995.

[40] B. T. Stander und J. C. Hart, „Guaranteeing the Topology of an Implicit Surface Polygonization for Interactive Modeling“, in *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques*, New York, NY, USA, 1997, S. 279–286.

[41] K. van Overveld und B. Wyvill, „Shrinkwrap: An efficient adaptive algorithm for triangulating an iso-surface“, *Vis. Comput.*, Bd. 20, Nr. 6, S. 362–379, Juli 2003.

[42] G. Taubin, „Curve and Surface Smoothing Without Shrinkage“, in *Proceedings of the Fifth International Conference on Computer Vision*, Washington, DC, USA, 1995, S. 852–.

[43] A. Opalach und S. Maddock, „Implicit surfaces: Appearance, blending and consistency“, in *In Fourth Eurographics Workshop on Animation and Simulation*, 1993, S. 233–245.

[44] P. Saalfeld, A. Baer, K. Lawonn, U. Preim, und B. Preim, „Das 3D User Interface zSpace“, in

Bildverarbeitung für die Medizin 2015, H. Handels, T. M. Deserno, H.-P. Meinzer, und T. Tolxdorff, Hrsg. Springer Berlin Heidelberg, 2015, S. 83–88.

[45] D. Hix und H. R. Hartson, *Developing User Interfaces: Ensuring Usability Through Product & Process*. New York, NY, USA: John Wiley & Sons, Inc., 1993.

[46] B. G. Witmer und M. J. Singer, „Measuring Presence in Virtual Environments: A Presence Questionnaire“, *Presence Teleoper Virtual Env.*, Bd. 7, Nr. 3, S. 225–240, Juni 1998.

[47] Oeltze, Steffen, „Visualisierung baumartiger anatomischer Strukturen mit Convolution Surfaces“, Diplomarbeit, Otto-von-Guericke Universität, Magdeburg, 2004.

[48] M.-P. Cani und S. Hornus, „Subdivision-curve primitives: a new solution for interactive implicit modeling“, in *Shape Modeling and Applications, SMI 2001 International Conference on.*, 2001, S. 82–88.

[49] H. Lee, S. Kwon, und S. Lee, „Real-time Pencil Rendering“, in *Proceedings of the 4th International Symposium on Non-photorealistic Animation and Rendering*, New York, NY, USA, 2006, S. 37–45.

[50] J. R. Siewert und H. J. Stein, Hrsg., *Chirurgie*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012.

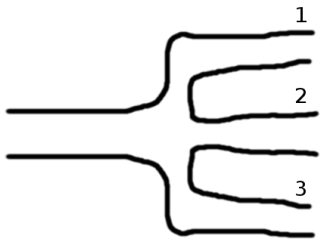
A Anhang

Im Anhang befinden sich:

- Aufgabenblatt der Evaluierung
- Fragebogen der Evaluierung

Aufgaben

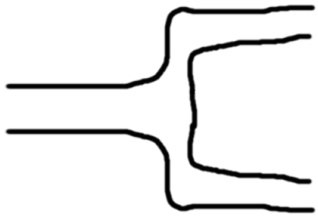
1a. Zeichnen Sie mit dem Stylus diese Trifurkation in die Szene.



- Ast 1 und 3 sollen dabei parallel zur Tischplatte und Ast 2 in Richtung Ihres Gesichtes gezeichnet werden.

1b. Aktivieren Sie „Hohe Meshqualität“ und betrachten Sie das Ergebnis aus verschiedenen Blickwinkeln.

1c. Wechseln Sie in den Editieren-Modus und löschen Sie Ast 2.



1d. Erstellen Sie in Ast 1 dieses Aneurysma.

1e. Erstellen Sie in Ast 3 diese Stenose.

2a. Löschen Sie alles mit der „Reset“-Box und zeichnen Sie nun ein gerades Blutgefäß in die Szene.

2b. Nutzen Sie die „Oversketching“-Funktion und ändern Sie die Form des Gefäßes, sodass es in etwa so aussieht.

2c. Zeichnen Sie an einer beliebigen Stelle des Gefäßes dieses Aneurysma.

Fragebogen

Auswertung der Daten erfolgt anonym

Wie alt sind Sie?		Jahre
Ihr Geschlecht?		
Welchen Beruf / Studiengang?		

	Ja	Nein
Haben Sie Erfahrungen mit Ray-based Interaktionen (Stiftinteraktionen im 3D Raum?)	☐	☐
Haben Sie Erfahrungen mit Grafiktabletts?	☐	☐
Haben Sie Probleme mit 3D-Sehen?	☐	☐

Teil 1

Die Software	- - -	- -	-	- / +	+	++	+++	*Die Software*
bietet einen schlechten Überblick über ihr Funktionsangebot.	☐	☐	☐	☐	☐	☐	☐	bietet einen guten Überblick über ihr Funktionsangebot.
erfordert überflüssige Eingaben.	☐	☐	☐	☐	☐	☐	☐	erfordert keine überflüssigen Eingaben.
erfordert bei Fehlern im großen und ganzen einen hohen Korrekturaufwand.	☐	☐	☐	☐	☐	☐	☐	erfordert bei Fehlern im großen und ganzen einen geringen Korrekturaufwand.
erfordert viel Zeit zum Erlernen.	☐	☐	☐	☐	☐	☐	☐	erfordert wenig Zeit zum Erlernen.
erfordert, dass man sich viele Details merken muss.	☐	☐	☐	☐	☐	☐	☐	erfordert nicht, dass man sich viele Details merken muss.
ermöglicht keinen leichten Wechsel zwischen einzelnen Menüs oder Masken.	☐	☐	☐	☐	☐	☐	☐	ermöglicht einen leichten Wechsel zwischen einzelnen Menüs oder Masken.

erschwert die Orientierung, durch eine uneinheitliche Gestaltung.	☐	☐	☐	☐	☐	☐	☐	erleichtert die Orientierung, durch eine einheitliche Gestaltung.
erzwingt eine unnötig starre Einhaltung von Bearbeitungsschritten.	☐	☐	☐	☐	☐	☐	☐	erzwingt keine unnötig starre Einhaltung von Bearbeitungsschritten.
erzwingt unnötige Unterbrechungen der Arbeit.	☐	☐	☐	☐	☐	☐	☐	erzwingt keine unnötigen Unterbrechungen der Arbeit.
informiert in unzureichendem Maße über das, was sie gerade macht.	☐	☐	☐	☐	☐	☐	☐	informiert in ausreichendem Maße über das, was sie gerade macht.
informiert zu spät über fehlerhafte Eingaben.	☐	☐	☐	☐	☐	☐	☐	informiert sofort über fehlerhafte Eingaben.
ist kompliziert zu bedienen.	☐	☐	☐	☐	☐	☐	☐	ist unkompliziert zu bedienen.
ist schlecht auf die Anforderungen der Arbeit zugeschnitten.	☐	☐	☐	☐	☐	☐	☐	ist gut auf die Anforderungen der Arbeit zugeschnitten.

ist schlecht ohne fremde Hilfe oder Handbuch erlernbar.	☐	☐	☐	☐	☐	☐	☐	ist gut ohne fremde Hilfe oder Handbuch erlernbar.
ist so gestaltet, dass kleine Fehler schwerwiegende Folgen haben können.	☐	☐	☐	☐	☐	☐	☐	ist so gestaltet, dass kleine Fehler keine schwerwiegenden Folgen haben können.
ist so gestaltet, dass sich einmal Gelerntes schlecht einprägt.	☐	☐	☐	☐	☐	☐	☐	ist so gestaltet, dass sich einmal Gelerntes gut einprägt.
lässt einen im Unklaren darüber, ob eine Eingabe erfolgreich war oder nicht.	☐	☐	☐	☐	☐	☐	☐	lässt einen nicht im Unklaren darüber, ob eine Eingabe erfolgreich war oder nicht.
lässt sich nicht durchgehend nach einem einheitlichen Prinzip bedienen.	☐	☐	☐	☐	☐	☐	☐	lässt sich durchgehend nach einem einheitlichen Prinzip bedienen.
liefert in unzureichendem Maße Informationen darüber, welche Eingaben zulässig oder nötig sind.	☐	☐	☐	☐	☐	☐	☐	liefert in zureichendem Maße Informationen darüber, welche Eingaben zulässig oder nötig sind.
liefert schlecht verständliche Fehlermeldungen.	☐	☐	☐	☐	☐	☐	☐	liefert gut verständliche Fehlermeldungen.

reagiert mit schwer vorhersehbaren Bearbeitungszeiten.	☐	☐	☐	☐	☐	☐	☐	reagiert mit gut vorhersehbaren Bearbeitungszeiten.
verwendet schlecht verständliche Begriffe, Bezeichnungen, Abkürzungen oder Symbole in Masken und Menüs.	☐	☐	☐	☐	☐	☐	☐	verwendet gut verständliche Begriffe, Bezeichnungen, Abkürzungen oder Symbole in Masken und Menüs.

Teil 2

1. Wie sehr waren Sie in der Lage, Ereignisse zu kontrollieren.

Gar nicht			Einigermaßen			Sehr
☐	☐	☐	☐	☐	☐	☐

2. Wie empfänglich war die Umgebung auf ihre Aktionen?

Gar nicht empfänglich			Einigermaßen empfänglich			Sehr Empfänglich
☐	☐	☐	☐	☐	☐	☐

3. Wie sehr haben Sie die Interaktionen mit den Blutgefäßen als natürlich empfunden?

Sehr unnatürlich			Grenzwertig			Sehr natürlich
☐	☐	☐	☐	☐	☐	☐

4. Waren Sie in der Lage vorherzusehen, was als nächstes als Antwort auf Ihre Eingabe passieren würde?

Gar nicht			Einigermaßen			Sehr
☐	☐	☐	☐	☐	☐	☐

5. Wie genau konnten Sie das Blutgefäß begutachten?

Gar nicht genau			Einigermaßen genau			Sehr genau
☐	☐	☐	☐	☐	☐	☐

6. Wie gut konnten Sie das Blutgefäß aus mehreren Blickwinkeln betrachten?

Gar Nicht			Einigermaßen			Ausgiebig
☐	☐	☐	☐	☐	☐	☐

7. Wie gut konnten Sie das Blutgefäß in der virtuellen Umgebung manipulieren?

Gar nicht			Einigermaßen			Ausgiebig
☐	☐	☐	☐	☐	☐	☐

8. Wie verwirrend war der Eingabemechanismus?

Gar nicht verwirrend			Einigermaßen verwirrend			Sehr verwirrend
☐	☐	☐	☐	☐	☐	☐

9. Wie viel Verzögerung gab es zwischen Ihren Aktionen und den erwarteten Ergebnissen?

Keine Verzögerung			Etwas Verzögerung			Lange Verzögerung
☐	☐	☐	☐	☐	☐	☐

10. Wie kompetent in der Bewegung und Interaktion haben Sie sich nach dem Ende der Nutzung gefühlt?

Gar nicht kompetent			Einigermaßen kompetent			Sehr kompetent
☐	☐	☐	☐	☐	☐	☐

11. Wie sehr hat das Eingabegerät Sie von der Durchführung Ihrer Aufgabe abgehalten?

Gar nicht			Einigermaßen			Sehr
☐	☐	☐	☐	☐	☐	☐

12. Wie sehr finden Sie, dass die Bedienelemente rechts im 3D Interface (Boxen, Labels) vergleichbar sind zu einem klassischen Interface?

Gar nicht vergleichbar			Einigermaßen vergleichbar			Sehr vergleichbar
☐	☐	☐	☐	☐	☐	☐