

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



Individualprojekt - Abschlussbericht

Verwendung impliziter Oberflächen zur Gefäß- und Flussvisualisierung für die medizinische Ausbildung

Aleksandar Stojnic

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



Selbstständigkeitserklärung

Hiermit versichere ich, dass ich den vorliegenden Abschlussbericht selbstständig und nur unter Zuhilfenahme der angegebenen Quellen erstellt habe.

Magdeburg, 02.10.2016


Aleksandar Stojnic

Zusammenfassung

Im Fokus dieses Individualprojekts stand die Erweiterung und Verbesserung eines zuvor entwickelten Prototyps zum semi-immersiven Zeichnen und Editieren von Blutgefäßen und Gefäßkrankheiten in 3D. Das Ziel hierbei war, mit Hilfe des Prototyps, Medizinstudierende in der Ausbildung zu unterstützen. Dieser Bericht beschreibt Konzept und Implementierung dieser Erweiterungen, zu denen Partikel-basierte Blutflusssimulation, Kugelselektion und Laden von Gefäßdatensätzen zählen. Zusätzlich wurde ein Großteil der Mängel, die in der Evaluierung des Prototyps erwähnt wurden, beseitigt.

Inhaltsverzeichnis

1	Einleitung	8
2	Grundlagen und Verwandte Arbeiten	10
2.1	Selektionsmethoden in 3D User Interfaces	10
2.1.1	Herausforderungen und Lösungsansätze bei Selektionen in 3D UIs	10
2.1.2	Virtual Hand-Metapher	12
2.1.3	Virtual Pointing-Metapher	12
2.2	Blutflusssimulation	14
2.2.1	Lagrange-basierte Simulation	15
3	Vorstellung des Prototyps	17
3.1	3D User Interface	17
3.2	Skizzieren	18
3.2.1	Zeichnen	18
3.2.2	Editieren	20
3.3	Erzeugung und Darstellung der Blutgefäße	21
3.4	Evaluierungsergebnisse	22
4	Neue Funktionen und Verbesserungen	24
4.1	Partikel-basierte Blutflusssimulation	24
4.2	Laden von Gefäßdatensätzen	26
4.3	Kugelselektion	27
4.4	Verschieben und Rotieren der Blutgefäßstruktur	29
4.5	Verbesserungen beim Skizzieren	30
4.5.1	Zeichnen	30
4.5.2	Editieren	30
4.5.3	Darstellung	31
4.6	Aortenbogen – Erstellung eines echten Gefäßsystems	31
5	Implementierung	35
5.1	Partikel-basierte Blutflusssimulation	35
5.1.1	Graphstruktur und Darstellung der Partikel	35

5.1.2	Partikelpfadsystem	36
5.1.3	Zuweisung der Pfade und Bewegen der Partikel	39
5.2	Kugelselektion	40
5.3	Verschieben der Blutgefäßstruktur	40
6	Zusammenfassung und Ausblick	42
6.1	Zusammenfassung	42
6.2	Ausblick	42
	Literaturverzeichnis	44

1 Einleitung

Gefäßkrankheiten, wie Aneurysmen und Stenosen, verursachen ca. 31% aller Todesfälle weltweit [1]. In dieser Hinsicht ist es sehr wichtig, dass zukünftige Ärzte in dem Bereich ausgebildet werden, damit diese Krankheiten rechtzeitig erkannt und behandelt werden können. Hierfür sind Kenntnisse und Verständnis über Blutgefäßstrukturen mit ihrer räumlichen Anordnung, sowie über Erkrankungen mit Krankheitsverlauf und ihren Folgen notwendig.

Bei der medizinischen Ausbildung werden dafür unter anderem Skizzen verwendet, die auf Tafeln reproduziert werden [2, 3] (siehe Abbildung 1). Dabei folgen Erklärungen mit schrittweiser Ergänzung neuer Details. Dadurch lassen sich komplexe Zusammenhänge gut illustrieren, allerdings ist die Qualität abhängig vom künstlerischen Talent des Zeichners [4]. Zudem sind die Skizzen aufwendig zu produzieren und können nur einen Blickwinkel darstellen. Dadurch können wichtige Informationen fehlen, da das Ergebnis darüber hinaus nicht wie ein echtes Gefäß aussieht. Weiterhin sind Skizzen nicht in der Lage, kontinuierlichen Blutfluss darzustellen. Dieses Wissen kann jedoch nützlich sein, um die Auswirkungen von Gefäßkrankheiten, wie Aneurysmen, zu verstehen [5].

Abbildung 1: Kreidezeichnungen von Blutgefäßstrukturen um umliegende Organe und Knochen (Bildquelle (a): [2], Bildquelle (b): [3]).

Diese Probleme können mit Hilfe von Software-Lösungen umgangen werden. Sie können darauf ausgelegt werden, die Blutgefäße in 3D darzustellen, wodurch sie realer wirken. Zudem erlauben sie die Betrachtung aus mehreren Blickwinkeln.

In der Bachelorarbeit von Stojnic [6] wurde eine Anwendung entwickelt, mit der der Nutzer Blutgefäße im 3D-Raum zeichnen und editieren, sowie Gefäßkrankheiten darstellen kann (siehe Abbildung 2). Hierfür wird das 3D User Interface zSpace verwendet. Dies ist ein stereoskopischer Bildschirm mit einem Stylus, welches sechs Freiheitsgrade (DOF) besitzt und somit Interaktionen in 3D ermöglicht. Die qualitative Evaluierung von Stojnic hat gezeigt, dass das zSpace und die verwendeten Interaktionstechniken sich eignen, um Blutgefäße im Raum zu zeichnen und zu editieren. Jedoch war die Anwendung anfällig auf Fehler und es fehlten wichtige Funktionen, wie das Verschieben und Rotieren von Blutgefäßstrukturen.

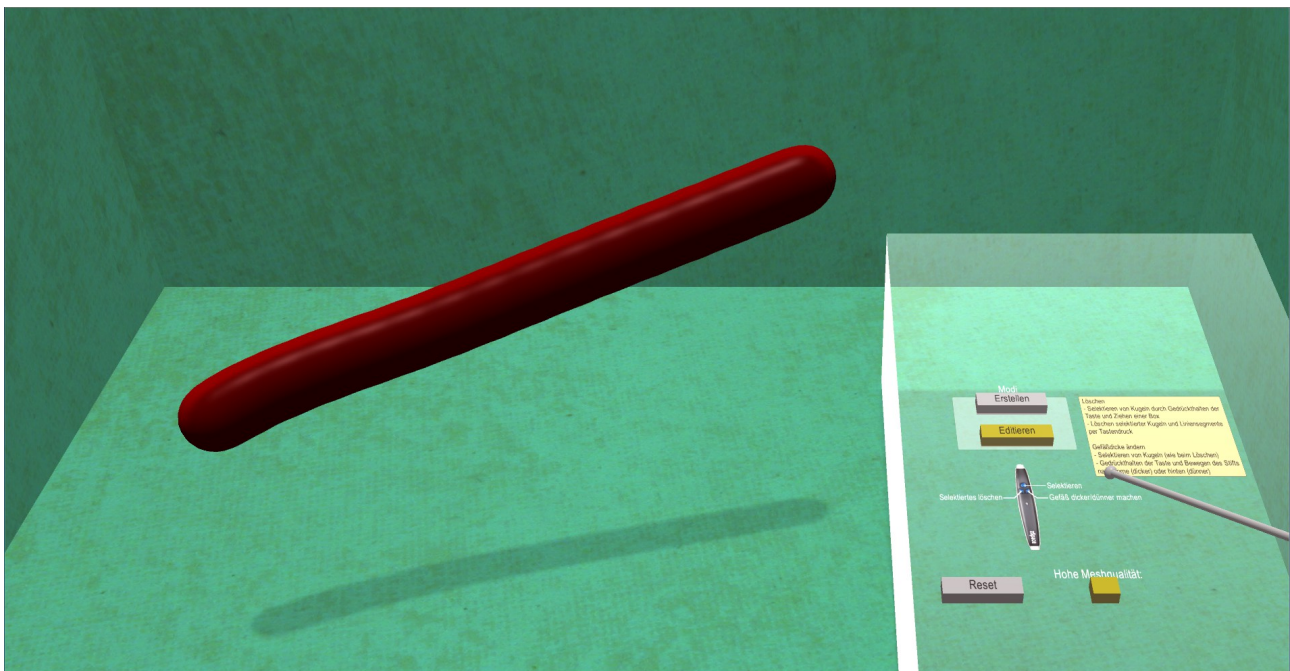


Abbildung 2: Die von Stojnic entwickelte Anwendung mit gezeichnetem Blutgefäß und Benutzungsoberfläche (Bildquelle: nach [6]).

Das Ziel dieses Individualprojekts war, die Anwendung so zu optimieren, dass sie zur Unterstützung für die medizinische Ausbildung verwendet werden kann. Medizinstudierende könnten somit selbstständig komplexe Blutgefäßstrukturen zeichnen und betrachten, die im Vergleich zu Kreidezeichnungen viel schneller zu produzieren sind. Darüber hinaus können neben der Darstellung Zusatzinformationen, wie z. B. Blutfluss, präsentiert werden. Zusammen mit der Möglichkeit, Stenosen und Aneurysmen einzufügen, können die Auswirkungen dieser Krankheiten beobachtet werden. Somit kann ein Medizinstudent leichter und schneller komplexe Zusammenhänge verstehen lernen.

Hierfür wurden im Verlauf des Individualprojekts, Mängel in der Anwendung der Bachelorarbeit beseitigt und neue Funktionen hinzugefügt. Im Fokus standen hierbei die Implementierung von Blutflusssimulation, das Laden von Gefäßdatensätzen und die Anpassung der Selektionsmethode.

Der Bericht ist folgendermaßen gegliedert:

Im zweiten Kapitel werden die Grundlagen von Selektionstechniken in 3D User Interfaces und von Blutflusssimulation erläutert. Es wird zudem auf verwandte Arbeiten eingegangen. Daraufhin wird im dritten Kapitel die Anwendung mit ihren Features vorgestellt. Anschließend werden im vierten Kapitel die Neuerungen und Verbesserungen erläutert. Im fünften Kapitel folgen darauf die Details der Implementierung. Zum Schluss wird im sechsten Kapitel eine Zusammenfassung und einen Ausblick darauf gegeben, was verbessert werden kann.

2 Grundlagen und Verwandte Arbeiten

In diesem Kapitel wird auf die Grundlagen von Selektionsmethoden in 3D User Interfaces (3D UI) eingegangen, wobei erklärt wird, welche Herausforderungen es beim Design von Selektionstechniken gibt und wie sie sich unterteilen lassen.

Anschließend folgen Grundlagen zur Blutflusssimulation, wobei hier spezifisch auf das Lagrange-Verfahren eingegangen wird, da diese für die Anwendung verwendet wird.

2.1 Selektionsmethoden in 3D User Interfaces

Die folgenden Erklärungen zu Selektionsmethoden in 3D User Interfaces basieren auf den Arbeiten von Argelaguet und Andujar [7] und Preim und Dachsel [8].

2.1.1 Herausforderungen und Lösungsansätze bei Selektionen in 3D UIs

Die Selektion von Objekten ist eine der fundamentalsten Aufgaben in 3D UI Anwendungen [9], denn sie ist oft die Aktion, die vor der Manipulation eines Objektes geschieht. Wenn die Methode schlecht konzeptioniert und inadequat für eine spezifische Situation ist, dann hat dies negative Auswirkungen auf die Leistungsfähigkeit des Nutzers.

Im Allgemeinen muss eine 3D Selektionsmethode folgende Anforderungen erfüllen. Sie muss:

- schnell,
- präzise,
- robust gegenüber Fehler vom Nutzer,
- leicht zu verstehen und zu kontrollieren und
- wenig ermüdend für die Hand sein.

Der letzte Punkt ist hierbei besonders wichtig, denn je nach Eingabegerät erfolgen Selektionen frei im Raum, ohne das die Hand irgendeine Stütze hat [10]. Mit dem *Optimized Initial Impuls Model* [11] lässt sich hierbei die Aktion in zwei Phasen unterteilen. In der Ersten erfolgt eine schnelle und unpräzise Bewegung zum Ziel und in der Zweiten erfolgen iterative und langsame Korrekturbewegungen. Dies kann auf Dauer ermüdend für die Hand sein.

Wie viel Bewegung letztendlich erforderlich ist, hängt von der Anzahl der Freiheitsgraden (eng. (engl. Degrees of Freedom) ab. In der Regel sind drei (Position) bis sechs (Position+Orientierung) DOF erforderlich, wobei dies von der gewählten Technik abhängt.

Es kann zudem bei einer Selektion im 3D-Raum das Problem auftreten, dass die Szene viele Objekte enthält, die dicht beieinander sind. Dadurch kann es bei Methoden, die ein Selektionsvolumen besitzen, vorkommen, dass mehrere Objekte gleichzeitig selektiert werden [12]. In diesem Fall sind zusätzliche Techniken erforderlich, um solche Mehrdeutigkeiten aufzulösen. Diese werden im Folgenden vorgestellt:

Manuelle Verfahren

Hier entscheidet der Nutzer selbst, welches Objekt am Ende selektiert wird. Zum Beispiel werden alle berührten Objekte in einer Liste oder einem Tortenmenü angezeigt, so wie beim Flow Ray von Grossman und Balakrishnan [13]. Danach wählt der Nutzer das gewünschte Objekt aus. Solche Verfahren erfordern zwar eine höhere kognitive Leistung, aber dafür sind sie flexibel und robust gegenüber Fehlern.

Heuristiken

Alle Objekte bekommen bei Berührung mit dem Selektionsvolumen eine Rangordnung nach einer gewissen Heuristik, z. B. wie nah sie an der zentralen Achse des Selektionszylinders sind [14]. Es wird dann das Objekt mit dem höchsten Rang ausgewählt. Die kognitive Leistung ist zwar geringer, jedoch ist die Selektion nicht immer präzise.

Auswahl nach Verhalten

Bei diesem Verfahren werden statistische Informationen über das Selektionsverhalten des Nutzers gesammelt, die einen Einfluss darauf haben, welches Objekt selektiert wird. Hierbei werden die Objekte kontinuierlich eingestuft nach Informationen, wie Zeit innerhalb des Selektionsvolumens, Nähe zum Zentrum, oder Anzahl der Berührungen mit dem Volumen. Werden z. B. zwei Objekte berührt und das zweite Objekt wurde über die Zeit öfter berührt als das Erste, dann wird das Zweite selektiert. IntenSelect [15] ist eine Selektionsmethode, die solch ein Verfahren verwendet. Auch hier ist weniger kognitive Leistung notwendig, aber Präzision kann nicht immer gewährleistet werden.

Die Präzision bei der Selektion kann aber auch durch Rauschen beim Eingabegerät oder durch Zittern der Hand beeinträchtigt werden. Um dies zu unterbinden, kann das sogenannte Control-Display-Ratio (CD-Ratio) angepasst werden. Diese bestimmt, wie sehr die Bewegungen des Eingabegerätes auf das virtuelle Selektionswerkzeug übertragen werden. Ist der Wert eins, dann stimmen die beiden Bewegungen überein. Dies wird als isomorphisches Mapping bezeichnet. Bei kleineren oder größeren Werten, wird die Bewegung des virtuellen Selektionswerkzeuges skaliert. Dies wird anisomorphisches Mapping genannt. Die CD-Ratio kann entweder manuell vom Nutzer, oder automatisch, wenn sich der Cursor nahe am Ziel befindet, angepasst werden. Die Einstellung kann aber auch nach der Geschwindigkeit, mit der das Eingabegerät bewegt wird, erfolgen.

Unabhängig davon, welche Methoden zur Reduktion von Mehrdeutigkeiten und zur Anpassung der CD-Ratio genommen werden, Feedback ist bei der Selektion essenziell, um den Nutzer Informationen über Objektzustände zu vermitteln. Vorwiegend wird hierbei visuelles Feedback, wie die Änderung der Form des virtuellen Cursors, oder visuelle Hervorhebung des selektierten Objektes, verwendet. Ergänzend dazu kann aber auch akustisches und haptisches Feedback hinzugenommen werden. Allerdings sollte mit Bedacht damit umgegangen werden, da zu häufige Töne und Vibrationen als störend empfunden werden können.

Bei der Auswahl einer geeigneten Selektionsmethode haben sich zwei Metaphern herausgebildet, die im folgenden näher betrachtet werden. Diese sind die Virtual Hand- und Virtual Pointing-Metapher.

2.1.2 Virtual Hand-Metapher

Bei der Virtual Hand [16] handelt es sich um einen virtuellen Cursor, der die Hand im virtuellen Raum repräsentiert. Der Cursor kann hierbei verschiedenen Formen haben. Am meisten verwendet werden Volumen, wie Zylinder, Quader, oder Kugeln.

Bei Virtual Hand-Methoden wird die 3D-Position der Hand getrackt. Die Selektion erfolgt dann durch Navigation und anschließendes Berühren des Objektes. Ein Beispiel für solch eine Methode ist der Bubble Cursor von Vanacken et al. [12]. Hier wird eine semi-transparente Kugel verwendet, deren Größe sich dynamisch verändern kann (siehe Abbildung 3). Um Mehrdeutigkeiten zu vermeiden, wird immer das Objekt selektiert, dass die geringste Distanz zum Zentrum der Kugel hat, welches mit einem Fadenkreuz gekennzeichnet ist. Das selektierte Objekt wird farbig markiert. Der Bubble Cursor bietet zudem eine Funktion, um verdeckte Objekte sichtbar zu machen. Alle Objekte, die sich in der Nähe des Cursors befinden, werden transparent dargestellt. Diese Funktion ist allerdings nur dann nützlich, wenn die ungefähren Positionen der Objekte bereits bekannt sind.

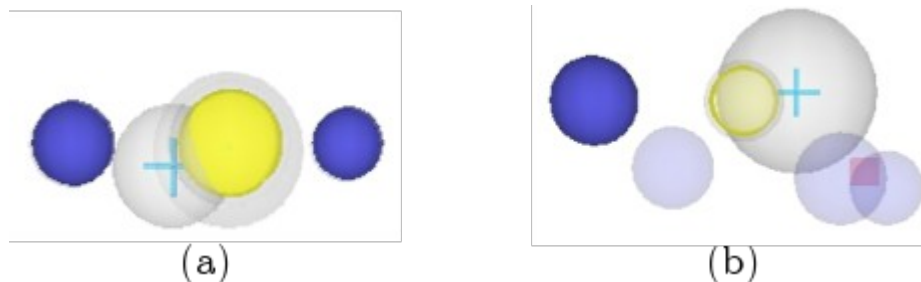


Abbildung 3: In (a) wird ein Objekt (blau) mit dem Bubble Cursor (graue Kugel mit Fadenkreuz) selektiert. Das Objekt wird darauf gelb markiert. In (b) ist zu sehen, wie der Bubble Cursor die naheliegenden Objekte transparent macht (Bildquelle: nach [12]).

Der Vorteil bei Virtual Hand-Methoden ist, dass Aktionen ähnlich zu Aktionen in der realen Welt sind und somit natürlich wirken. Allerdings sind Objekte nicht selektierbar, wenn sie außerhalb der Reichweite des Armes sind. In solchen Fällen ist es oftmals günstiger, andere Selektionsmethoden zu verwenden, um die Einschränkungen des menschlichen Körpers zu überwinden.

2.1.3 Virtual Pointing-Metapher

Bei dieser Methode erfolgt die Selektion durch einen Strahl, der meist vom Eingabegerät aus in die Unendlichkeit geschossen wird (Raycast) [17]. Das Objekt, dass vom Strahl berührt wird, wird selektiert.

Virtual Pointing benötigt lediglich die Orientierung des Handgelenks. Die Rotation ist hierbei nicht notwendig, weshalb 2-DOF zur Steuerung ausreichen. Somit ist weniger physikalische Bewegung notwendig als bei der Virtual Hand. Zudem lassen sich damit Objekte selektieren, die sehr weit entfernt sind. Sind diese aber sehr klein, dann ist das Treffen mit dem Strahl schwierig, da hierfür viel Präzision nötig ist.

Um dies zu umgehen, gibt es einige Ansätze. Steinicke et al. [18] nutzen neben dem normalen Raycast einen zweiten Strahl, der sich zum nächstgelegenen Objekt hinbiegt (siehe Abbildung 4). Dieser Strahl bleibt an dem Objekt kleben, selbst wenn die Hand ein wenig bewegt wird. So wird

vermieden, dass ständig zwischen zwei nahe gelegenen Objekten hin und her gewechselt wird. Nachteil allerdings ist, dass der Nutzer nicht in der Lage ist, mehrere Objekte gleichzeitig zu selektieren.

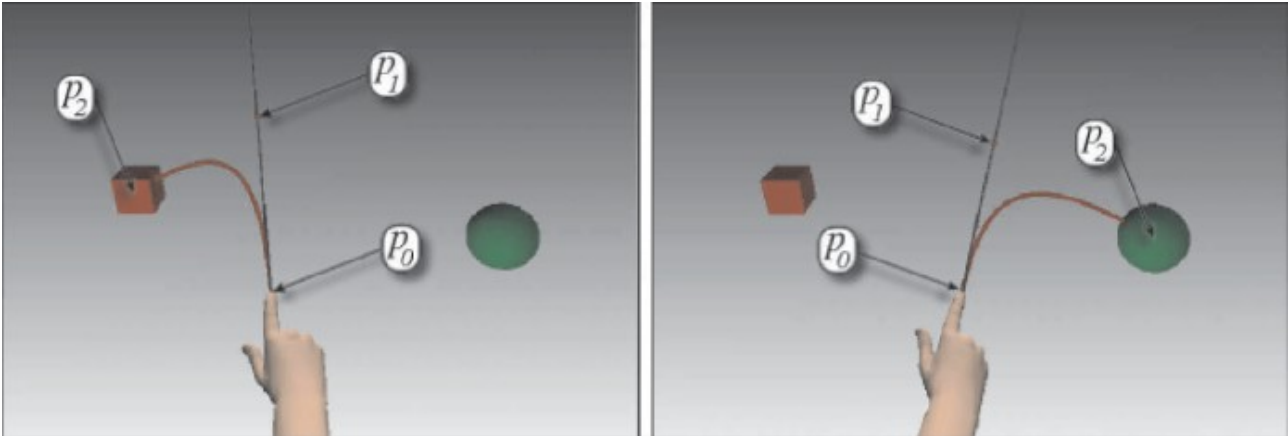


Abbildung 4: Darstellung des zweiten, gebogenen Strahls von Steinicke et al., der sich zum nächstgelegenen Objekt hinbiegt (Bildquelle: [18]).

Alternativ kann statt einem Strahl ein Kegel in die Unendlichkeit geschossen werden, in der alle Objekte innerhalb des Volumens selektiert werden. Somit lässt sich zwar das Selektieren kleiner weit entfernter Objekte lösen, allerdings treten dann wieder Mehrdeutigkeiten auf. Steed und Parker haben hierfür die Shadow Cone Selektion entwickelt [19]. Bei dieser Gruppenselektionstechnik erfolgt, wie in Abbildung 5 zu sehen, zuerst eine initiale Selektion mit dem Kegel. Anschließend wird er in eine neue Position gebracht. Es werden dann nur die Objekte selektiert, die sich bis zum Ende der Bewegung innerhalb des Kegels befinden. Evaluierungen haben gezeigt, dass dadurch weniger Fehler auftreten.

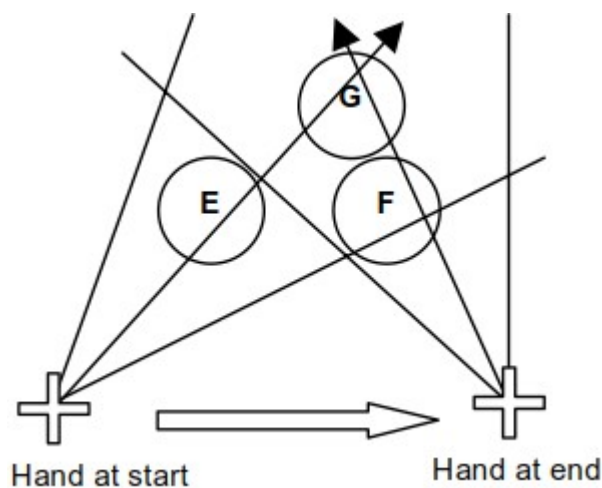


Abbildung 5: Darstellung von Steed und Parkers Shadow Cone Selektion. Am Anfang befinden sich Objekt E und G vollständig im Kegel. Nach Bewegung der Hand sind E und F vollständig enthalten. Da G bis zum Ende den Kegel nicht verlassen hat, wird G selektiert (Bildquelle: nach [19]).

Der Vorgang wird jedoch schwierig, wenn sich viele Objekte dicht nebeneinander befinden. In solchen Situationen können Mehrdeutigkeiten auftreten, die mit bloßer Bewegung des Kegels schwer zu unterbinden sind. Hierfür wären entsprechende Methoden zur Auflösung solcher Mehrdeutigkeiten notwendig.

Kopper et al. [20] nutzen Raycast in einer Supermarkt-Szene und projizieren auf der getroffenen Fläche einen Kreis (siehe Abbildung 6 (a)). Der Radius ist hierbei höher, je weiter entfernt die getroffene Fläche ist. Da nun immer mehrere Objekte selektiert werden, werden sogenannte Quad-Menüs benutzt, um Mehrdeutigkeiten aufzulösen (siehe Abbildung 6 (b)). Hierbei werden die Objekte gleichmäßig in vier Quadranten dargestellt. Der Nutzer wählt dann den aus, der das gewünschte Objekt enthält. Der Quadrant wird anschließend erneut unterteilt. Die Selektion wird somit so lange progressiv verfeinert, bis nur noch das gewünschte Objekt übrig bleibt. Diese Selektionsmethode funktioniert gut bei Objekten, die in einer Fläche angeordnet sind, so wie bei der Supermarkt-Szene, und wenn sich das Zielobjekt visuell von den anderen abhebt.

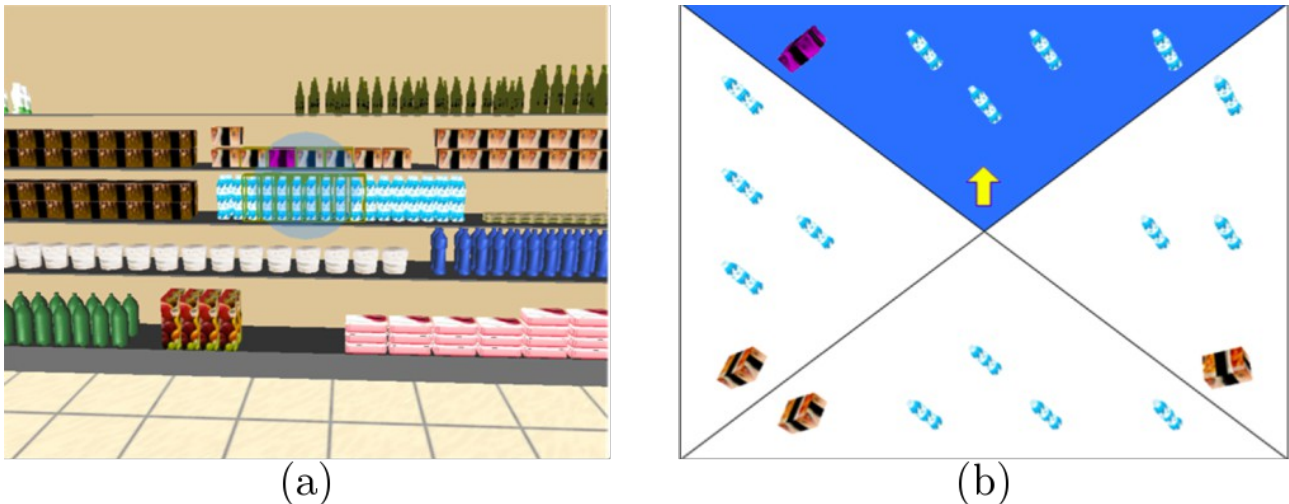


Abbildung 6: In (a) werden alle Objekte innerhalb des Kreises selektiert und anschließend in (b) in einem Quad-Menü dargestellt. Der Nutzer wählt dann den Quadranten aus, der das gewünschte Objekt enthält (Bildquelle: nach [20]).

Ob nun Virtual Hand oder Virtual Pointing Methoden verwendet werden, hängt von der Anwendung ab und wie die Objekte selektiert werden sollen. Argelaguet und Andujar [7] präsentieren eine tabellarische Darstellung verschiedener Selektionstechniken und klassifizieren sie nach verschiedenen Eigenschaften, wie DOF, CD-Ratio und Selektionsvolumen. Diese soll Entwicklern helfen, die für sie geeignetste Selektionsmethode auszuwählen.

2.2 Blutflusssimulation

Die folgenden Erklärungen zur Blutflusssimulation basieren auf den Arbeiten von Braley und Sandu [21] und von Liu und Liu [22].

Um Fluide, zu denen auch Blut gehört, zu simulieren, gibt es zwei Ansätze. Der eine Ansatz wird Euler-basierte Simulation genannt. Hier werden im Volumen, durch das die Flüssigkeit durchfließt, fixe Punkte über die Zeit hinweg gemessen. Diese Punkte sind hierbei oft in einem Gitter angeordnet (siehe Abbildung 7 (b)).

Das Problem bei diesem Verfahren ist, dass das Gitter auch Bereiche umfassen kann, die keine Flüssigkeit enthalten, da gezeichnete Blutgefäße nur wenig Platz einnehmen. Das führt zu unnötigem Rechenaufwand und somit zu einer potentiellen Reduktion der Framerate. Da es aber

für die Anwendung wichtig ist, dass die Interaktivität auch bei laufender Simulation erhalten bleibt, wird der zweite Ansatz, die sogenannte Lagrange-basierte Simulation, verwendet.

2.2.1 Lagrange-basierte Simulation

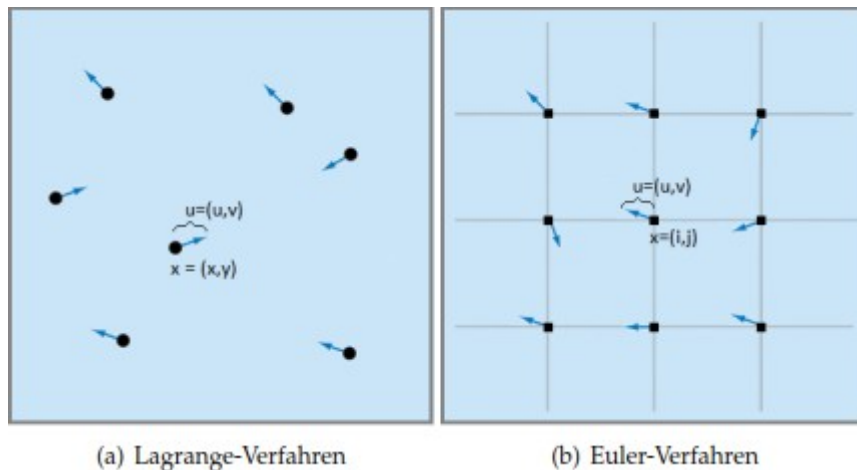


Abbildung 7: Darstellung, an welchen Punkten die Parameter gemessen werden. Beim Lagrange-Verfahren werden diese bei den Partikeln und beim Euler-Verfahren an festen Gitterpunkten gemessen (Bildquelle: [23]).

Bei diesem Ansatz wird die Flüssigkeit durch Partikel simuliert, die über verschiedene Eigenschaften zur Beschreibung von Fluiden, wie Masse, Dichte oder Geschwindigkeit, verfügen (siehe Abbildung 7 (a)). Lagrange-basierte Verfahren haben den Vorteil, dass nur die Partikel und somit nur die Flüssigkeit für die Simulation betrachtet werden, statt wie bei Euler alle Gitterpunkte. Dadurch entsteht weniger Rechenaufwand, wodurch dieses Verfahren besser geeignet ist, um die Interaktivität der Anwendung zu gewährleisten.

Ein Beispiel für ein Lagrange-basiertes Verfahren ist die geglättete Teilchen-Hydrodynamik (engl. Smoothed Particle Hydrodynamics). Hier erfolgt die Simulation mit Hilfe von Gleichungen der klassischen Newton'schen Hydrodynamik. Dabei werden für jedes Partikel die Parameter

- Position,
- Geschwindigkeit,
- Masse,
- Dichte und
- Druck bestimmt.

Gesteuert wird dies durch eine Glättungsfunktion. Die Parameter werden approximiert durch Summation der Werte der Nearest-Neighbor Partikel.

Abbildung 8: Darstellung der Partikel in einem Gefäß im System von Müller et al. Die Farbe visualisiert hierbei die Geschwindigkeit. Schnelle Partikel sind gelb. Langsame Partikel sind rot (Bildquelle: [5]).

Geglättete Teilchen-Hydrodynamik bei der Blutflusssimulation wird in der Arbeit von Müller et al. [5] verwendet (siehe Abbildung 8). Die Parameter wurden hierbei direkt aus den Navier-Stokes Gleichungen abgeleitet. Zudem wurden an der Gefäßwand Pseudo-Partikel platziert, die sich mit den Partikeln der Flüssigkeiten abstoßen. Dies simuliert Interaktionen mit der Gefäßwand, da sie sich dadurch leicht verformt. Qin et al. [24] erweiterte diese Wandinteraktion später um eine Implementierung des Masse-Feder Modells, um zusätzlich das Rupturieren der Gefäßwände bei Aneurysmen (siehe Abbildung 9) zu simulieren.

Abbildung 9: Darstellung des Rupturierens eines Aneurysmas im System von Qin et al. (Bildquelle: [24]).

Für das Individualprojekt wurde eine reale Simulation des Blutflusses nicht implementiert. Stattdessen wird den Partikeln die Richtung, in der sie sich entlangbewegen sollen, zugewiesen. Zwar ist diese Art der Simulation weniger realistisch, aber sie reicht aus, um eine plausible Darstellung des Blutflussverhaltens zu illustrieren. Zudem ist diese Variante schnell und leicht zu implementieren.

3 Vorstellung des Prototyps

In diesem Kapitel wird der von Stojnic [6] entwickelte Prototyp, welcher mit der Game Engine Unity entwickelt wurde, mit all seinen Funktionen zusammenfassend vorgestellt. Es wird darauf eingegangen, wie mit der 3D UI Linien gezeichnet und editiert und wie die Blutgefäße erstellt werden.

3.1 3D User Interface

Der Prototyp verwendet das 3D User Interface zSpace (siehe Abbildung 10). Dies ist ein stereoskopischer Bildschirm, der von zSpace Inc. entwickelt wurde [25]. Zu seinen Komponenten zählt eine passive 3D Brille mit Markern, die von zwei Infrarot-Kameras am Bildschirm getrackt werden. Dies ermöglicht die Bewegungsparallaxe, mit der Objekte per Kopfbewegungen aus verschiedenen Blickwinkeln betrachtet werden können. Zudem besitzt das zSpace einen Stylus mit 6-DOF, der ebenfalls von den Infrarot-Kameras getrackt wird. Dies erlaubt freies Zeichnen von Blutgefäßen im 3D-Raum.



Abbildung 10: zSpace 100 Modell mit Brille und Stylus (Bildquelle: techbriefs¹).

Dies erfolgt durch die drei Aktionstasten am Stylus und durch die Verwendung von Modi. Es gibt einen Erstellen-Modus zum Zeichnen von Blutgefäßen und einen Editieren-Modus zum Löschen und zur Änderung der Gefäßdicke. Der Stylus wird zudem virtuell in die Szene verlängert, damit der Nutzer mit den Objekten interagieren kann.

Unten rechts in der Anwendung befinden sich Bedienelemente aus 3D-Boxen und Beschriftungen, die von einem Glaskasten umschlossen werden (siehe Abbildung 11). Die 3D-Boxen funktionieren analog zu Buttons in konventionellen Benutzungsschnittstellen und dienen zum Wechsel zwischen Modi, der Wiederherstellung des Ausgangszustandes und zur Veränderung der Qualität des Oberflächenmodells. Die Beschriftungen dienen zur Darstellung von Anweisungen für die einzelnen Aktionen und zum Zeigen der Tastenbelegung.

1 http://www.techbriefs.com/images/stories/NTB/2012/FEATURES/40431-155_fig2.png, zuletzt aufgerufen: 04.09.2016

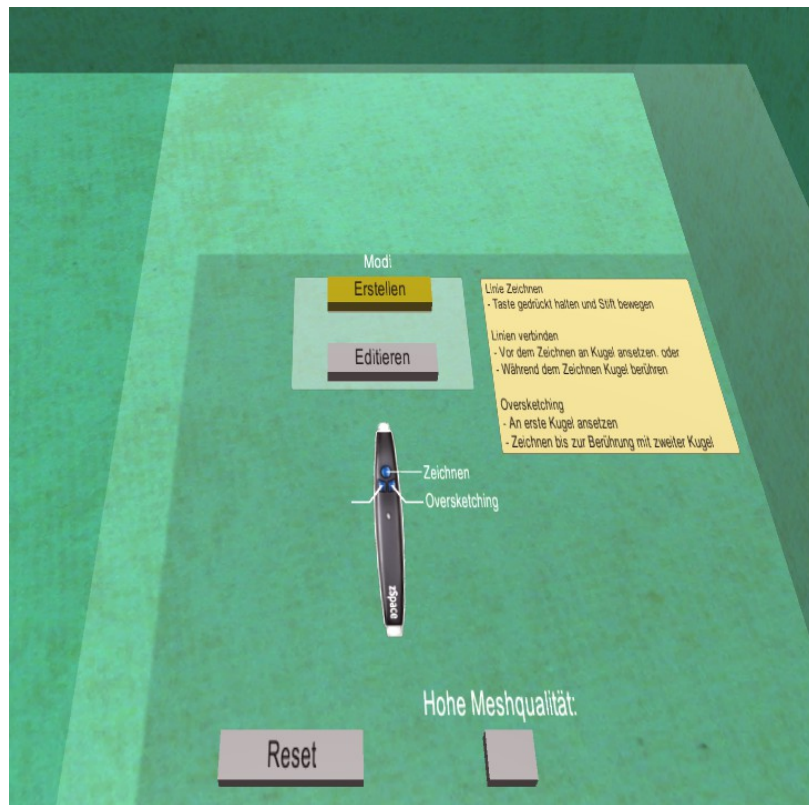


Abbildung 11: Bedienelemente der Anwendung (Bildquelle: [6]).

3.2 Skizzieren

3.2.1 Zeichnen

Die Eingaben des Stylus werden als Positionsinformationen in einem 3D-Koordinatensystem interpretiert, die in irregulären Zeitabständen abgetastet werden. Eine Linie ist somit eine zeitabhängige Sequenz von 3D Punkten.

Der Nutzer zeichnet die Mittellinie des Gefäßes und das Oberflächenmodell wird um diese Linie generiert. An jedem abgetasteten Punkt wird dabei eine Kugel (Kontrollpunkt) platziert. Um Ungenauigkeiten, wie z. B. Zittern der Hand, zu unterbinden, wird während des Zeichnens ein Resampling (siehe Abbildung 12) und eine 5×1 Gauß-Filterung angewendet. Das Resampling sorgt dafür, dass die Kugeln den gleichen Abstand zueinander haben. Der Gauß-Filter dient zur Glättung von Kurven.

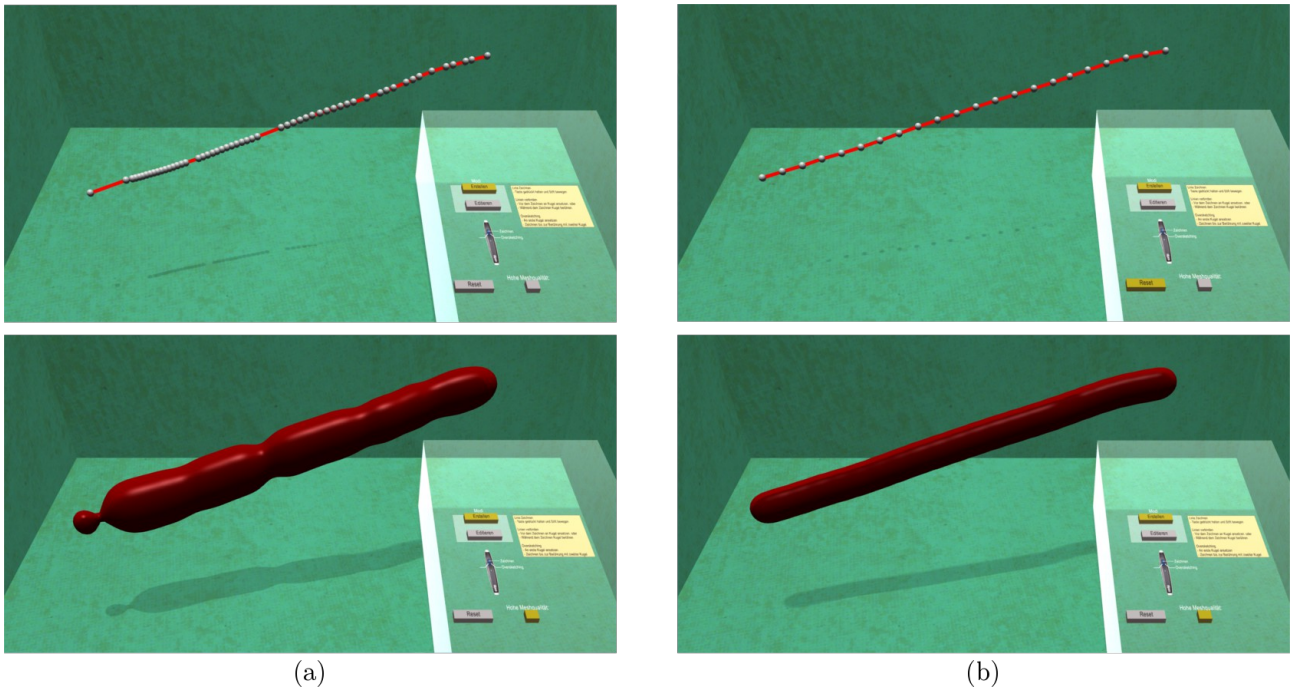


Abbildung 12: Die Mittellinie des Gefäßes in (a) ohne und in (b) mit Resampling. Darunter sind die daraus entstandenen Gefäße dargestellt (Bildquelle: [6]).

Um Linien miteinander zu verbinden, setzt der Nutzer entweder mit der Spitze des Stylus an einen Kontrollpunkt an, oder berührt einen während des Zeichnens (siehe Abbildung 13).

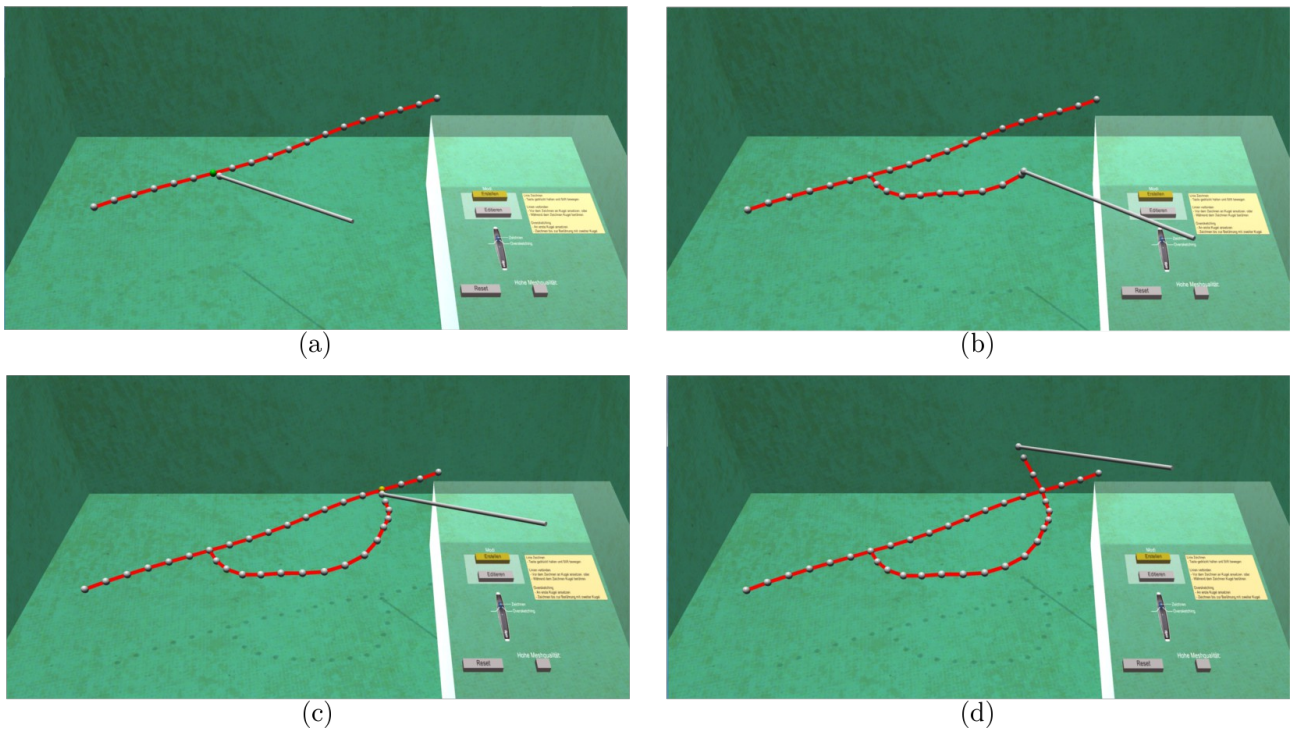


Abbildung 13: Darstellung des Prozesses beim Verbinden von Linien (Bildquelle: [6]).

3.2.2 Editieren

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

Selektieren und Löschen

Um Kontrollpunkte zu selektieren, kann der Nutzer per Tastendruck im Editieren-Modus eine halbtransparente Box aufziehen, dessen Größe durch Bewegung des Stylus beeinflussbar ist. Kontrollpunkte, die sich innerhalb der Box befinden, werden selektiert und ändern entsprechend ihre Farbe. Diese Methode entspricht der Rubberband-Metapher [26] in 3D.

Drückt der Nutzer nun auf die entsprechende Taste, werden die selektierten Kontrollpunkte mit den entsprechenden Liniensegmenten gelöscht.

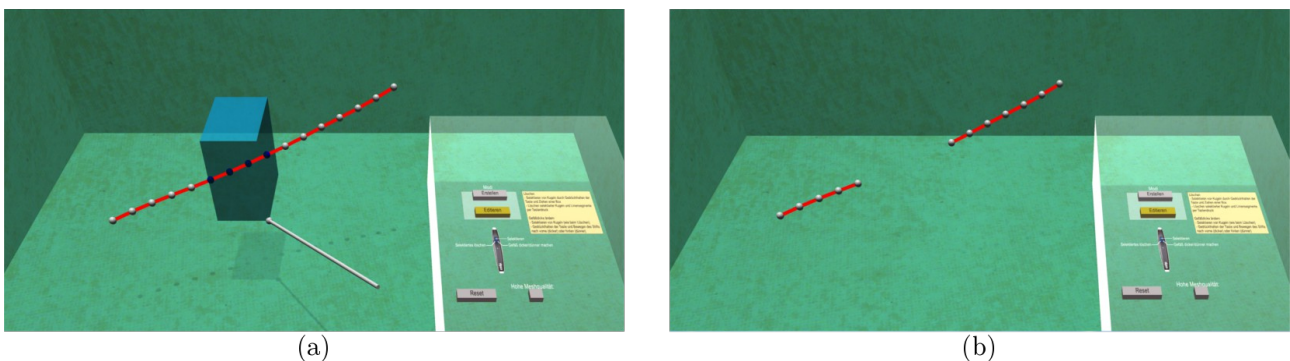


Abbildung 14: Darstellung des Selektierens von Kontrollpunkten (a) mit Löschen (b) (Bildquelle: [6]).

Neuzeichnen (Oversketching)

Es ist möglich, den Verlauf eines Liniensegmentes nach dem Zeichnen zu ändern. Dazu setzt der Nutzer an einen Kontrollpunkt an, hält die entsprechende Taste gedrückt und zeichnet eine neue Linie, bis die Stylusspitze einen zweiten Kontrollpunkt berührt. Die Kontrollpunkte, die sich zwischen den Berührten befinden, werden anschließend durch die Kontrollpunkte der neuen Linie ersetzt. Dies funktioniert allerdings nur, wenn sich kein Verbindungspunkt dazwischen befindet. Ansonsten können Verbindungen gelöst werden.

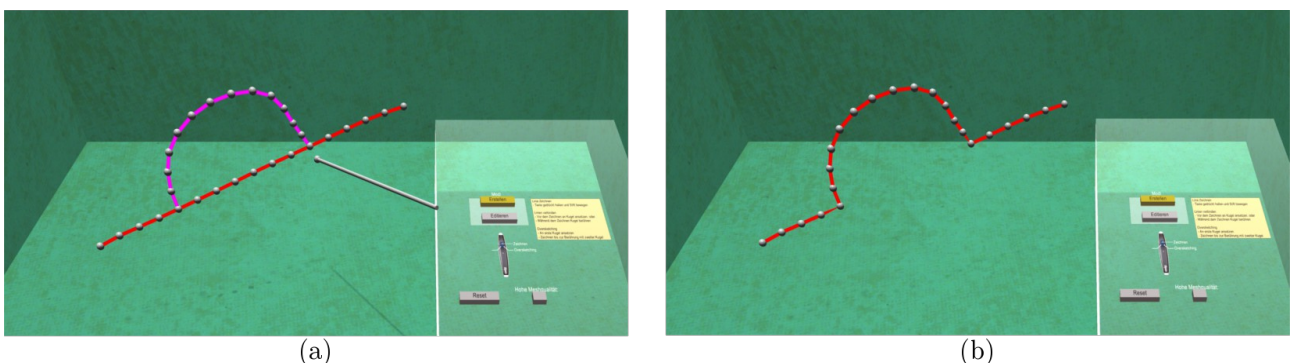


Abbildung 15: Oversketching der roten Linie durch die gezeichnete magenta Linie (Bildquelle: [6]).

3.3 Erzeugung und Darstellung der Blutgefäße

Zur Erzeugung der Blutgefäße werden implizite Oberflächen verwendet. Eine implizite Oberfläche ist eine Funktion, die ein Skalarfeld bildet [27], welches anschließend, wie hier in der Anwendung, mit dem Marching Cubes Algorithmus [28] polygonisiert werden kann. Für den Prototyp werden Metaballs [29] verwendet, deren implizite Funktion aus Polynomen zweiten Grades besteht.

Eine besondere Eigenschaft von Metaballs ist das Blending. Werden zwei Kugeln zusammengebracht, verschmelzen sie zu einer einzigen Kugel mit glatten Übergängen (siehe Abbildung 16). Zudem kann der Radius eines Metaballs verändert werden, was sie geeignet macht, um Gefäßkrankheiten, wie Aneurysmen und Stenosen, darzustellen.

Abbildung 16: Darstellung des Blending zwischen zwei Kugeln (Bildquelle: nach [27]).

Um nun das Oberflächenmodell des Blutgefäßes zu erstellen, wird an jedem Kontrollpunkt ein Metaball platziert. Durch das Blending formen sie eine Röhrenstruktur. Ein Nachteil hierbei ist, dass die Gittergröße des Marching Cubes Algorithmus einen hohen Einfluss auf die Leistungsfähigkeit der Anwendung hat. Deshalb ist die Gittergröße beim Zeichnen niedrig eingestellt. Der Nutzer kann bei den Bedienelementen die Funktion „Hohe Meshqualität“ aktivieren, um ein besseres Ergebnis zu erhalten.

Um Stenosen oder Aneurysmen, wie in Abbildung 17 dargestellt, zu erstellen, muss der Nutzer zunächst im Editieren-Modus mindestens einen Kontrollpunkt selektieren. Durch Gedrückthalten der entsprechenden Taste und Bewegung des Stylus nach hinten wird das Gefäß dicker. Bewegt der Nutzer den Stylus nach vorne, wird das Gefäß dünner.

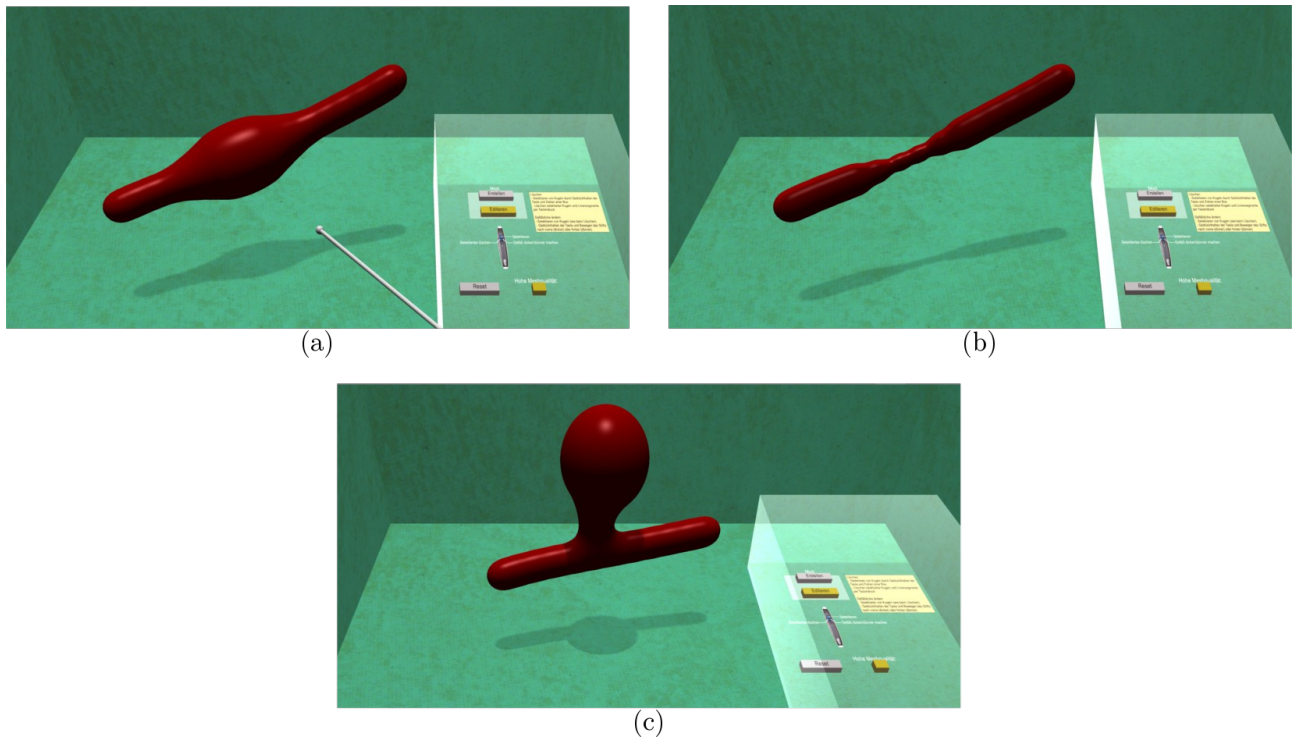


Abbildung 17: Möglichkeiten zur Erstellung von Gefäßkrankheiten. In (a) ist ein spindelförmiges Aneurysma, in (b) eine Stenose und in (c) ein sackförmiges Aneurysma dargestellt (Bildquelle: [6]).

Das Oberflächenmodell wird in einer dunkelroten Farbe dargestellt. Berührt die Stylusspitze das Modell, dann wird per Fresnel-Shading mit Front-Face Culling die Mittellinie sichtbar gemacht (siehe Abbildung 18).

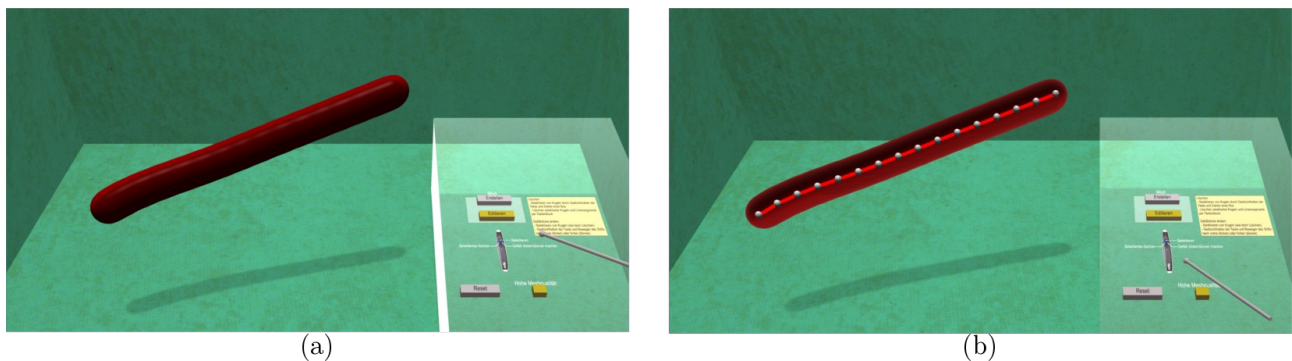


Abbildung 18: Blutgefäß in (a) mit opakem Shader und in (b) mit Fresnel-Shader (Bildquelle: [6]).

3.4 Evaluierungsergebnisse

Der Prototyp wurde qualitativ per Fragebögen und mit der Think-aloud Methode [30] evaluiert. Es gab insgesamt neun Probanden, darunter fünf Männer und vier Frauen. Dabei waren sieben von ihnen Computervisualistik-Studierende. Die anderen beiden waren jeweils ein Informatik-Student und ein Doktorand.

Alle Probanden empfanden das Zeichnen, Editieren und die Bedienelemente als intuitiv. Die Ergebnisse der Fragebögen waren zudem positiv. Bei dem 3D UI wurde von den Probanden ein

Nachlassen der Stereoskopie bemerkt, wenn von der Seite geschaut wurde.

Beim Zeichnen wurde das Verbinden von Linien als unintuitiv empfunden, da erwartet wurde, dass die Verbindung sofort bei der Berührung erfolgen würde (Snapping). Dies war aber nicht der Fall. Zudem wurde bei Fehlern oft von vorne angefangen, statt die Editiermöglichkeiten zu nutzen. Viele Probanden hatten angemerkt, dass es keine Möglichkeit gab, Gefäße zu rotieren oder zu verschieben, da die Bewegungsparallaxe allein nicht ausreichte, um alle Blickwinkel zu betrachten. Beim Editieren fanden die Probanden es schwierig einzuschätzen, wie weit sich die Selektionsbox in die Tiefe bewegte, da die Kanten nicht hervorgehoben waren. Zudem wurde kritisiert, dass die Änderung der Dicke zu schnell verlief und dass die Dicke beim erneuten Tastendruck auf den Ursprungswert zurückgesetzt wurde. Dadurch war es nicht möglich, Korrekturänderungen vorzunehmen.

Bei den Bedienelementen wurde oft beobachtet, dass die Probanden vergessen hatten, die Funktion „Hohe Meshqualität“ auszustellen, bevor sie wieder anfangen haben zu zeichnen.

Aus den Ergebnissen wurde geschlossen, dass das zSpace und die verwendeten Interaktionstechniken zum Zeichnen von Blutgefäßen und Gefäßkrankheiten geeignet sind. Allerdings ist durch die Limitationen der Bewegungsparallaxe das zSpace nur von einer Person effektiv nutzbar. Zudem machen Nutzer viele Fehler und nutzen die Editiermöglichkeiten nicht, um diese zu beseitigen, was bedeutet, dass diese alleine nicht ausreichen.

4 Neue Funktionen und Verbesserungen

In diesem Kapitel werden die neuen Funktionen und Verbesserungen, die im Verlauf des Individualprojekts implementiert wurden, vorgestellt. Zuerst wird auf die Konzepte der Partikel-basierten Blutflusssimulation und dem Laden von Gefäßdatensätzen eingegangen. Anschließend wird die neue Kugelselektion und das Verschieben und Rotieren der Blutgefäßstruktur beschrieben. Zum Schluss folgt eine Zusammenfassung der sonstigen Verbesserungen, die beim Zeichnen, Editieren und bei der Darstellung umgesetzt wurden.

4.1 Partikel-basierte Blutflusssimulation

Der Nutzer kann nun eine beliebige Anzahl an Blutflussquellen an einem Kontrollpunkt platzieren und bestimmen, in welche Richtung die Partikel durch das Gefäß entlangfließen (siehe Abbildung 19). Hierfür wurde der neue Blutfluss-Modus hinzugefügt. Um eine Blutflussquelle zu platzieren, setzt der Nutzer an einen Kontrollpunkt an. Durch Drücken der entsprechenden Taste, wird von dem Kontrollpunkt aus ein Pfeil aufgezo-gen. Wird nun ein benachbarter Kontrollpunkt berührt und dabei die Taste losgelassen, beginnen Partikel in Richtung des Pfeiles zu wandern. Berührte Kontrollpunkte werden hierbei zur Nutzerunterstützung farblich hervorgehoben. Um eine Blutflussquelle zu entfernen, setzt der Nutzer an einen Kontrollpunkt mit einer Quelle an, zieht einen neuen Pfeil auf und führt diese in einen freien Bereich, anstatt einen neuen Kontrollpunkt zu berühren.

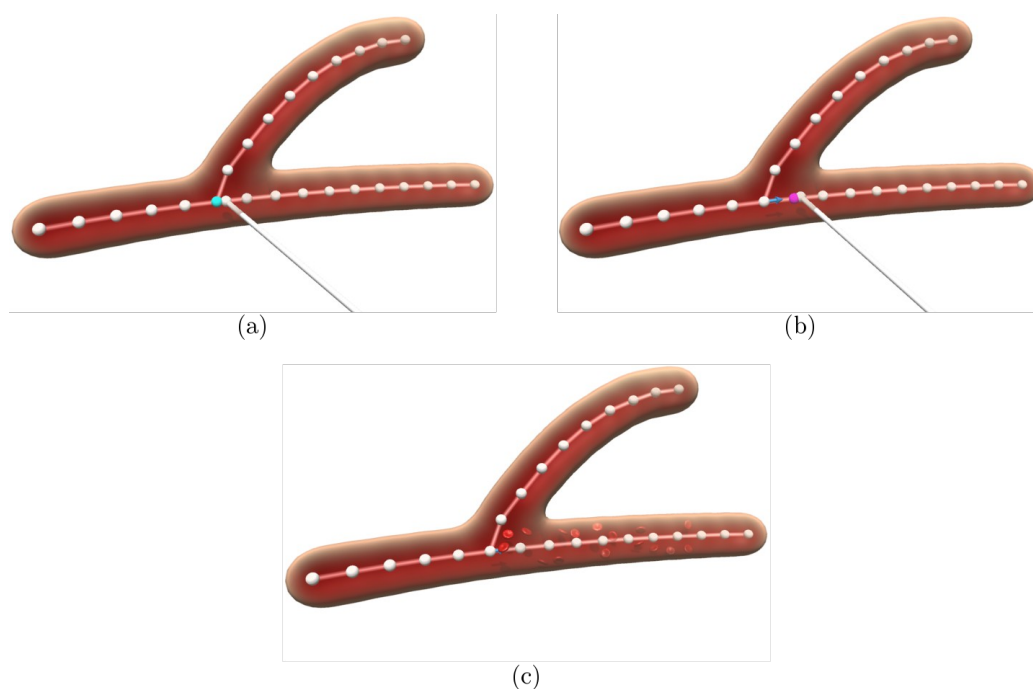


Abbildung 19: Platzieren einer Blutflussquelle. Es wird an einen Kontrollpunkt angesetzt (a) und ein Pfeil zu einem benachbarten Kontrollpunkt aufgezo-gen (b). Blutplättchen fließen dann in die angegebene Richtung (c).

Auf diese Weise kann der Nutzer Ort und Richtung des Blutflusses bestimmen. Die Darstellung des Pfeils dient hierbei als Feedback. Der Nutzer kann zudem dadurch die Simulation nur auf die Bereiche beschränken, die für ihn relevant sind. Das Platzieren erfolgt darüber hinaus wie beim Zeichnen und Editieren durch Interaktion mit den Kontrollpunkten, wodurch keine neuen Datenstrukturen oder Algorithmen implementiert und keine neuen Aktionen gelernt werden mussten.

Die Partikel werden als rote Blutplättchen visualisiert, da beim echten Blutfluss solche Zellen durch die Gefäße transportiert werden. Dadurch wirkt die Simulation authentisch. Die Blutplättchen folgen der Mittellinie des Gefäßes, bis sie das Ende eines Liniensegmentes erreichen und dort verschwinden. Erreichen sie dabei eine Verzweigung, wie z. B. eine Bifurkation, teilen sie sich entsprechend auf. Es fließt abwechselnd ein Partikel in die erste und ein Partikel in die zweite Richtung usw. (siehe Abschnitt 5.1.3).

Der Nutzer kann dabei die Form der Partikel durch Drücken der Nummerntasten 1, 2 und 3 ändern, falls Strukturen dargestellt werden, die sich für Blutplättchen wenig eignen, wie z. B. Bronchialbäume. Es sind die Formen

- Rote Blutplättchen (1),
- Lungenbläschen (2) und
- 2D-Bilder (3), die der Kamera zugewandt sind (Billboard-Sprites), möglich.

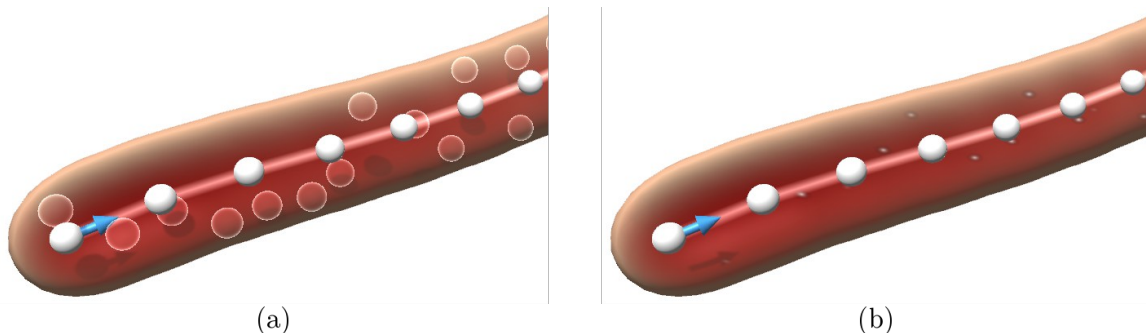


Abbildung 20: Die beiden anderen Darstellungsformen für Partikel. In (a) sind Lungenbläschen und in (b) sind Billboard-Sprites zu sehen.

Der Blutfluss ist zudem in der Lage, sich bei Änderungen der Gefäßstruktur anzupassen. Wenn während des Blutflusses gezeichnet, editiert, oder die Gefäßstruktur bewegt wird, dann werden alle Partikel gelöscht. Die Partikel, die danach erscheinen, folgen der neuen Struktur.

Durch diese Funktionalität kann ein Medizinstudent lernen, wie Blut durch eine Gefäßstruktur fließt. Ein großes Problem ist allerdings, dass die Simulation nicht dem realen Blutfluss im menschlichen Körper entspricht. Die Partikel folgen nicht den Regeln der Fluidodynamik. Es gibt lediglich eine Kollisionsabfrage, um zu verhindern, dass die Partikel nicht die Gefäßwand verlassen. Dies ist ein Nachteil bei der Darstellung von Aneurysmen und Stenosen, da dadurch die negativen Auswirkungen dieser Krankheiten nicht illustriert werden können. In solchen Fällen wäre ein Vergleich mit physikalisch basierten Simulationen sinnvoll, um abzuschätzen, wie viel Realismus notwendig ist, um diese Auswirkungen darzustellen.

Ein weiteres Problem ist, dass der Blutfluss bei Zyklen fehlerhaft ist. Partikel verschwinden nach einem Zyklus, da diese als Endpunkt des Liniensegments betrachtet werden. Darüber hinaus fließen Partikel nicht durch eine Verzweigung, wenn die Verzweigung sich zwischen einem Zyklus befindet (siehe Abbildung 21).

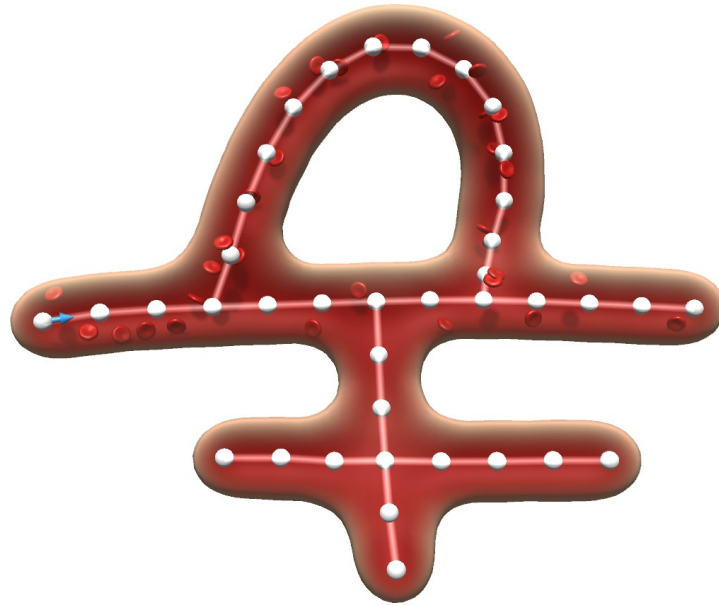


Abbildung 21: Das Problem bei Zyklen. Die Blutplättchen fließen nicht durch den Pfad zwischen Anfangs- und Endpunkt eines Zykluses.

Weiterhin hat der Nutzer keine Möglichkeit, die Richtung des Blutflusses nach Platzierung zu verändern. Stattdessen kann die Blutflussquelle entfernt und neu platziert, oder weitere Blutflussquellen gesetzt werden. Dadurch ergibt sich jedoch ein weiteres Problem. Wenn zu viele Partikel auf einmal sichtbar sind, dann nimmt die Framerate der Anwendung ab. Der Nutzer kann lediglich die Form der Partikel auf die Billboard-Sprites umstellen, um die Framerate zu verbessern.

4.2 Laden von Gefäßdatensätzen

Der Nutzer kann Gefäßdatensätze, wie z. B. den Circle of Willis, aus txt-Dateien in die Szene laden und diese anschließend betrachten (siehe Abbildung 22). Die Implementierung des Textparsers wurde hierbei von Patrick Saalfeld bereitgestellt.

Die Anwendung erstellt aus den Dateien die Mittellinie mit Kontrollpunkten. Hierbei ist an jedem Kontrollpunkt angegeben, wie groß der einzelne Metaball sein soll. Diese Gefäßstrukturen lassen sich dann, wie in Abschnitt 4.4 beschrieben, rotieren und verschieben, um sie aus unterschiedlichen Perspektiven zu betrachten.

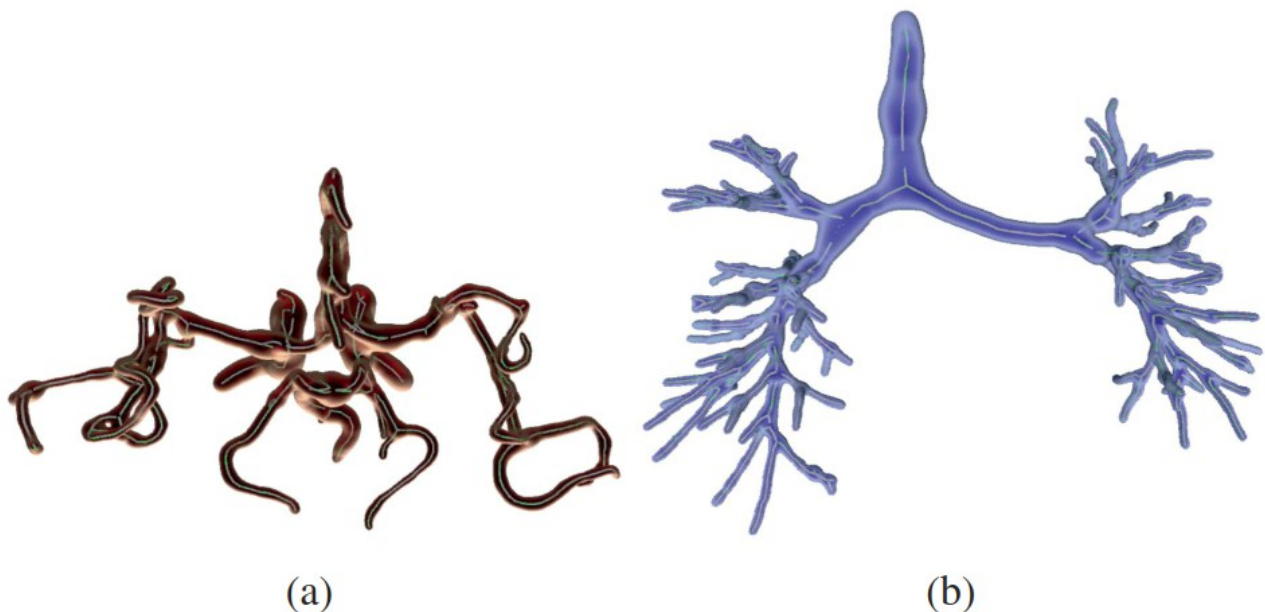


Abbildung 22: Beispiele für Gefäßdatensätze, die in die Anwendung geladen werden können. (a) ist der Circle of Willis und (b) ist ein Bronchialbaum (Bildquelle: [4]).

Der Medizinstudent kann dadurch den Aufbau und die räumliche Anordnung diverser Blutgefäßstrukturen im Körper verstehen lernen und dank der 3D Darstellung, können diese aus mehreren Blickwinkeln betrachtet werden.

Ein Problem ist jedoch, dass das Blutgefäßmodell nicht durch neue Strukturen per Zeichnen ergänzt werden kann. Die Kontrollpunkte sind zahlreich und oft eng aneinander angeordnet. Diese müssen sehr klein sein, sonst würden sie das Oberflächenmodell des Gefäßes überlagern. Es wird dadurch allerdings sehr schwer, einen bestimmten Kontrollpunkt zu treffen, um eine Verbindung herzustellen. Zudem ist das neu gezeichnete Gefäß oft viel zu dick und muss entsprechend angepasst werden. So ist die Wahrscheinlichkeit hoch, dass das Endergebnis nicht den Vorstellungen des Nutzers entspricht. Darüber hinaus erhöht sich bei der hohen Anzahl an Kontrollpunkten und somit Metaballs die Zeit, in der das Oberflächenmodell aufgebaut wird.

Weiterhin ist Blutfluss im geladenen Gefäß nicht möglich. Zum einen sind die Gefäße oft viel zu eng, um Blutplättchen durchzulassen, da deren Größe während der Simulation nicht veränderbar ist. Zum anderen hat die Mittellinie durch Ungenauigkeiten in den Datensätzen viele kleine Lücken, wodurch ein kontinuierlicher Blutfluss nicht gewährleistet werden kann. Zudem hat die Benutzungsoberfläche keine Bedienelemente, um Parameter des Gefäßmodells, wie z. B. die Größe, einzustellen. Diese müssen im Programmcode festgelegt werden.

4.3 Kugelselektion

Anstatt wie bisher eine transparente Box aufzuziehen, erscheint nun eine halbtransparente Kugel um die Spitze des Stylus, wenn in den Editieren-Modus gewechselt wird. Auf eine Virtual-Pointing Methode wurde verzichtet, da es zum einen mit einem Strahl schwer wäre, die Kontrollpunkte zu treffen, vor allem wenn mehrere hintereinander selektiert werden sollen. Zum anderen könnte es bei der Verwendung eines Kegels vorkommen, dass Strukturen hinter ausgewählten Kontrollpunkten

unabsichtlich selektiert werden. Deshalb wird eine Virtual Hand-Methode verwendet, da der Nutzer den Stylus beim Zeichnen umherbewegt. Auf diese Weise wären die Aktionen für Zeichnen und Selektion konsistent.

Die Kugel hat eine feste Größe, um sicherzustellen, dass immer nur ein Kontrollpunkt gleichzeitig selektiert werden kann. Zwar wäre es nützlich durch eine größere Kugel mehr auf einmal zu selektieren, jedoch könnten dabei aus Versehen umliegende Strukturen getroffen werden.

Per Druck auf die mittlere Taste wird die Kugel aktiv und ändert entsprechend ihre Farbe. Jeder Kontrollpunkt, der von der Kugel berührt wird, wird selektiert (siehe Abbildung 23). Um zu deselektieren, muss der Kontrollpunkt erneut berührt werden. Möchte der Nutzer die gesamte Selektion aufheben, muss die Taste kurz gedrückt und wieder losgelassen werden.

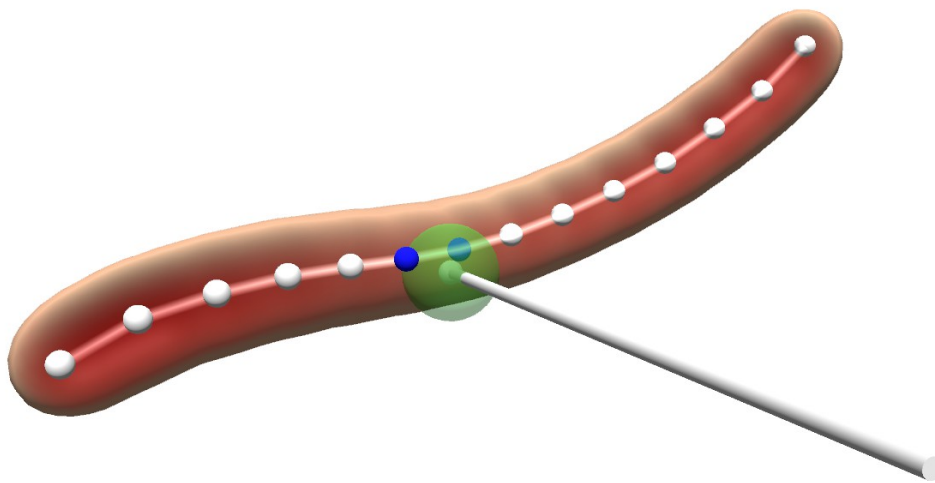


Abbildung 23: Darstellung des Selektionsprozesses mit der Kugel.

Mit der neuen Selektionsmethode ist es nun auch möglich, Kontrollpunkte sofort bei Berührung mit der Kugel zu löschen. Dafür muss der Nutzer nur die Taste für das Löschen gedrückt halten. Diese Aktion eignet sich für kleine und schnelle Änderungen. Dadurch braucht der Nutzer nicht vorher zu selektieren und spart dadurch Zeit. Dieses Prinzip funktioniert auch bei der Änderung der Gefäßdicke. Welche Aktion die Kugel durchführt, ist entsprechend mit der Farbe gekennzeichnet. Diese sind in Abbildung 24 zu sehen.

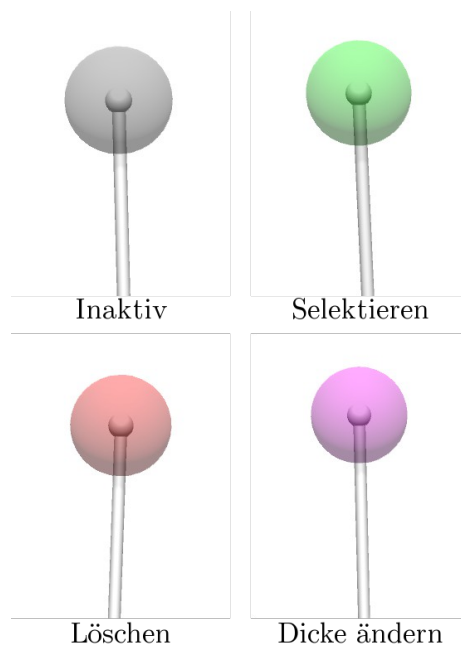


Abbildung 24: Die vier Farben, die den Zustand der Selektionskugel repräsentieren.

Ein Vorteil dieser Selektionsmethode ist, dass sie weniger fehleranfällig ist. Bei der Box bestand die Möglichkeit, dass Kontrollpunkte unbeabsichtigt selektiert werden konnten. Zudem ist die neue Methode schneller und flexibler. Der Nutzer muss lediglich über das Gefäß drüberfahren um einen großen Bereich zu selektieren. Darüber hinaus kann mehr als nur ein zusammenhängender Bereich selektiert werden. Bei der alten Methode musste auf die Position der Box geachtet werden, was den Prozess verlangsamte.

Da es sich hier um eine Virtual Hand-Technik handelt, besteht hier der Nachteil, dass Kontrollpunkte außerhalb der Reichweite des virtuellen Stylus nicht selektiert werden können. Zudem kann dies potenziell ermüdend für die Hand sein.

4.4 Verschieben und Rotieren der Blutgefäßstruktur

Durch Berühren eines Kontrollpunktes und Gedrückthalten der entsprechenden Taste, kann der Nutzer die Gefäßstruktur verschieben und rotieren (siehe Abbildung 25). Diese Aktion ist natürlich, da sie analog zum Greifen von realen Gegenständen funktioniert. Zudem kann der Nutzer das Gefäß in Blickwinkeln betrachten, die mit der Bewegungsparallaxe nicht erreichbar sind. Darüber hinaus können nun Liniensegmente selektiert werden, die vorher unerreichbar waren.

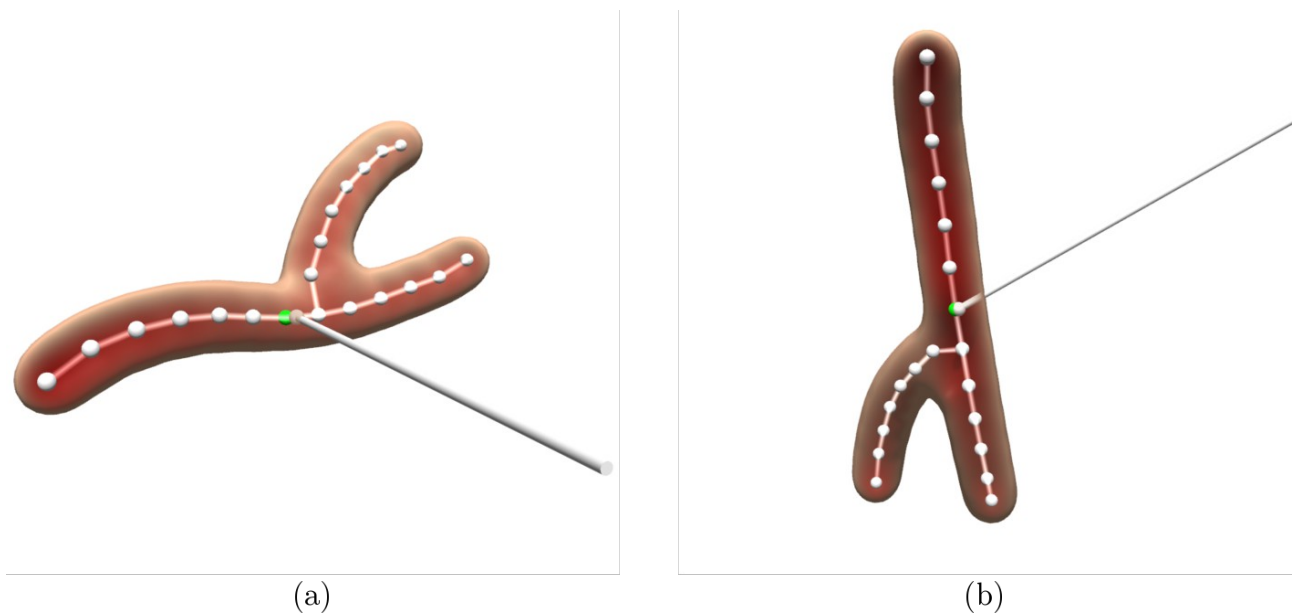


Abbildung 25: Verschieben und Rotieren des Gefäßes in eine Lage, die mit der Bewegungsparallaxe schwer erreichbar ist.

Jedoch besteht das Problem, dass zwei separate Gefäßstrukturen als eine Struktur interpretiert werden und somit beide verschoben werden (siehe Abschnitt 5.3). Weiterhin kann die Aktion ermüdend für das Handgelenk werden, da es zum Rotieren oft in unnatürliche Positionen gebracht werden muss.

4.5 Verbesserungen beim Skizzieren

4.5.1 Zeichnen

Beim Verbinden von Linien wurde Snapping implementiert. Wenn der Nutzer beim Zeichnen einen Kontrollpunkt berührt und anschließend die Taste loslässt, werden die Liniensegmente miteinander verbunden. Dadurch soll das Verbinden näher an den Erwartungen des Nutzers liegen, damit Verbindungsfehler, die bei der Evaluierung auftraten, vermieden werden.

4.5.2 Editieren

Bei der Veränderung der Blutgefäßdicke wurde die Skalierung angepasst, sodass die Veränderung nicht mehr ganz so schnell wie davor erfolgt. Dies erlaubt feinere Anpassungen. Weiterhin wird die Dicke nicht mehr auf den Standardwert zurückgesetzt, wenn erneut eine Veränderung vorgenommen wird. Somit sind anschließende Korrekturen möglich.

4.5.3 Darstellung

Die Visualisierung des Oberflächenmodells ändert sich nicht mehr bei Berührung mit der Stylusspitze. Es wird jetzt nur noch der Fresnel-Shader verwendet. Dieser wurde von Patrick Saalfeld so angepasst, dass mit der Einstellung der Fresnel-Stärke die Mittellinie komplett verdeckt werden kann. Dies geschieht nun, wenn die Funktion „Hohe Meshqualität“ aktiviert wird. Somit müssen die Nutzer es deaktivieren, wenn sie zeichnen wollen, denn sonst sind die Kontrollpunkte nicht sichtbar.

Weiterhin wurde von Patrick Saalfeld die Benutzungsoberfläche kosmetisch angepasst (siehe Abbildung 26). Die Farbe der Wände wurde von grün zu blauweiß geändert und die 3D-Boxen wurden durch abgekantete Boxen ersetzt. Das alles diente lediglich zur ästhetischen Verbesserung der Szene. Darüber hinaus wurden die deutschen Labels ins Englische übersetzt, da die Sprache von mehr Menschen auf der Welt gesprochen wird und somit mehr Leute die Anwendung nutzen können.

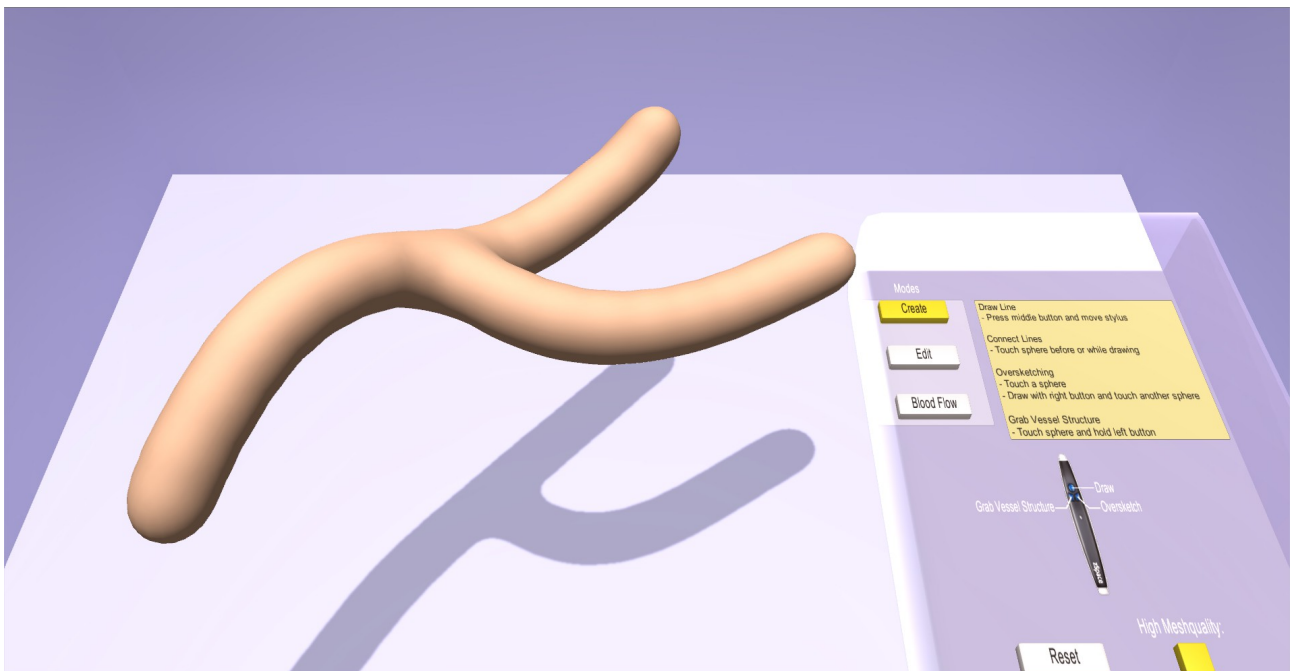


Abbildung 26: Das neue Aussehen der Anwendung.

4.6 Aortenbogen – Erstellung eines echten Gefäßsystems

Im Folgenden wird das Skizzieren des Aortenbogens (siehe Abbildung 27) im menschlichen Körper mit der Anwendung beschrieben und welche Schwierigkeiten dabei auftreten. Dies soll die Stärken und Schwächen der Anwendung illustrieren.

Der Aortenbogen ist ein Teil der Aorta, der großen Hauptschlagader, die aus dem Herzen entspringt und sich im Körper in viele kleinere Arterien aufspaltet [31]. Der Bogen verbindet die aufsteigende und absteigende Aorta miteinander und bildet den Ursprung für Arterienstämme, die zum Kopf, Hals und zu den oberen Extremitäten verlaufen.

Abbildung 27: Darstellung des Aortenbogens mit aufsteigender und absteigender Aorta, sowie ausgehenden Arterien (Bildquelle: wikipedia²).

Eine Möglichkeit, die in Abbildung 27 dargestellte Skizze in der Anwendung zu zeichnen, wäre zuerst den großen Bogen zu skizzieren und anschließend die kleineren Arterien hinzuzufügen (siehe Abbildung 28 (a) und (b)). Der Nutzer kann dann die Bewegungsparallaxe oder das Greifen nutzen, um die Struktur aus unterschiedlichen Perspektiven zu betrachten. Blutfluss kann während oder nach dem Zeichnen eingefügt werden (siehe Abbildung 28 (c)). Beide Möglichkeiten führen zum selben Ergebnis.

² https://de.wikipedia.org/wiki/Aortenbogen#/media/File:Gray506_de.svg, zuletzt aufgerufen: 24.09.2016

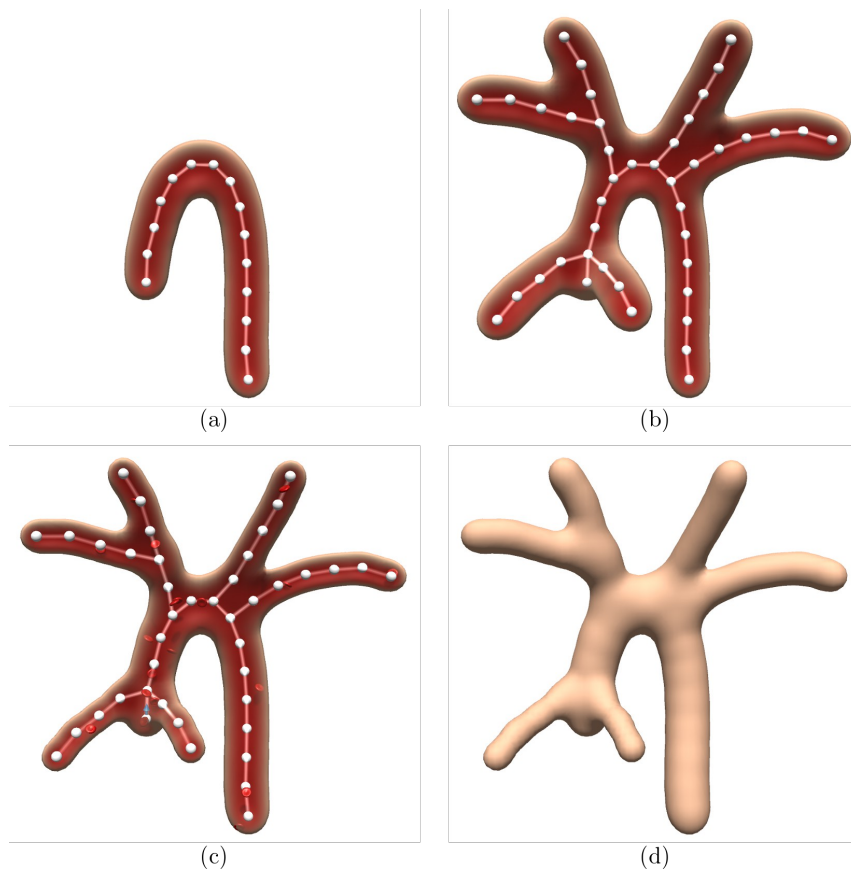


Abbildung 28: Skizzieren des Aortenbogens in der Anwendung. In (a) wird zuerst der Bogen und in (b) die dünneren Arterien gezeichnet. (c) zeigt den Blutfluss und (d) die fertige Struktur, in der die abzweigenden Arterien dünner gemacht wurden.

Die Struktur kann schnell und ohne Probleme skizziert werden. Wenn Fehler auftreten, dann lassen sich diese durch Löschen und Oversketching beheben. Der Nutzer kann zudem den Ursprung und den Verlauf einzelner Gefäße variieren. Dies ermöglicht, kleine anatomische Unterschiede zwischen unterschiedlichen Personen darzustellen. Allerdings muss dabei darauf geachtet werden, dass die Gefäße nicht zu nah aneinander gezeichnet werden, sonst können diese ungewollt miteinander verschmelzen. Dies ist vor allem beim Platzieren von Aneurysmen wichtig. Weiterhin kann es beim häufigen Oversketching vorkommen, dass die Mittellinie einen zackigen Verlauf bekommt. Das liegt daran, dass Start- und Endpunkt nach Ersetzen des Liniensegmentes nicht geglättet werden.

Das Zeichnen der Topologie der Gefäßstruktur ist einfach, jedoch treten Probleme auf, wenn die Dicke der dünneren Gefäßstämme angepasst werden soll. Wird ein Gefäß zu dünn, dann entstehen Wellenstrukturen, da dann die Abstände zwischen den Kontrollpunkten zu groß wird (siehe Abbildung 29). Es können zudem auch Diskontinuitäten auftreten, da für dünne Strukturen die Gittergröße des Marching Cubes zu groß ist. Zwar könnte dies durch eine kleinere Gittergröße und durch kleinere Abstände zwischen Kontrollpunkten behoben werden, jedoch würde damit der Aufbau des Oberflächenmodells länger dauern. Die Interaktivität wäre dann nicht mehr gewährleistet. Eine Möglichkeit, dieses Problem zu umgehen, wäre die gesamte Gefäßstruktur sehr dick zu machen, wodurch die Wahrscheinlichkeit für Diskontinuitäten niedriger wäre. Allerdings kann dabei der Effekt des ungewollten Blendings verstärkt werden.

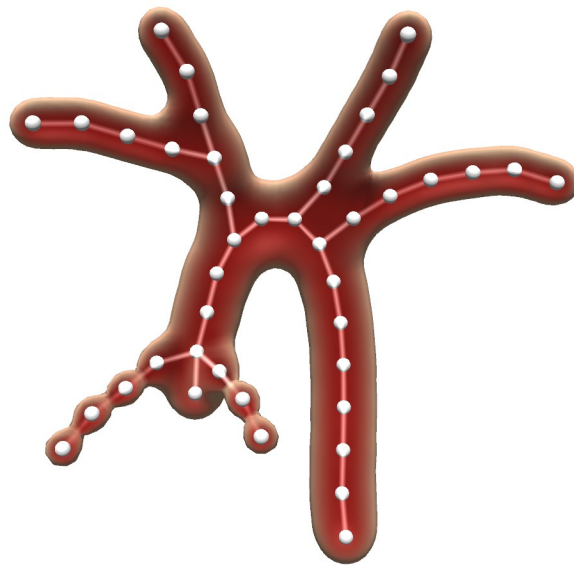


Abbildung 29: Darstellung der Wellenstruktur bei zu dünnem Metaball-Radius.

Dieses Beispiel zeigt, dass mit der Anwendung das Zeichnen von Topologien von Blutgefäßen schnell und einfach funktioniert. Jedoch kann das Nachbilden von unterschiedlichen Gefäßdicken unter Umständen zu degenerierten Gefäßen führen. Medizinstudierende können somit nur die Gefäßstrukturen zeichnen und betrachten, die keine sehr dünnen Gefäßstämme enthalten.

5 Implementierung

In diesem Kapitel wird beschrieben, wie die im vorigen Kapitel beschriebenen Änderungen mit der Game Engine Unity implementiert wurden. Im Fokus steht hierbei die Implementierung der Blutflusssimulation.

5.1 Partikel-basierte Blutflusssimulation

5.1.1 Graphstruktur und Darstellung der Partikel

Jeder Kontrollpunkt enthält eine Liste seiner Nachbarn mit denen er verbunden ist, sowie eine Pfadnummer. Diese Listen werden bei jeder Änderung der Topologie aktualisiert. Dadurch bilden die Kontrollpunkte einen ungerichteten Graphen. Mit Hilfe dieses Graphen und den Verbindungsinformationen der Kontrollpunkte können Pfade von der Blutflussquelle aus bis zu den Endpunkten von Liniensegmenten gebildet werden. Die Pfadnummer dient hierbei als Indikator, zu welchem der möglichen Pfade ein Kontrollpunkt gehört.

Zur Darstellung der Partikel wird Unitys Shuriken Partikelsystem verwendet. Dies ist eine Komponente, die an einem Kontrollpunkt-Gameobject angehängt wird, wenn der Nutzer eine Blutflussquelle platziert. Unity-Komponente³ wie diese stellen viele Funktionalitäten bereit, die Objekten zugeteilt werden können. Sie können unter anderem Position, Aussehen und Verhalten festlegen. GameObjects⁴ sind Conatainer für Komponenten.

Wie in Abbildung 30 zu sehen ist, besteht das Partikelsystem aus vielen einzelnen Modulen, die verschiedene Einstellungsmöglichkeiten für Partikel bieten. Diese lassen sich entweder im Unity Editor oder im Programmcode manipulieren. Für die Implementierung sind allerdings nur das Hauptmodul, Emission-, Shape-, Collision- und Renderer-Modul relevant.

³ <http://docs.unity3d.com/Manual/Components.html>, zuletzt aufgerufen 04.09.2016

⁴ <http://docs.unity3d.com/Manual/class-GameObject.html>, zuletzt aufgerufen 04.09.2016

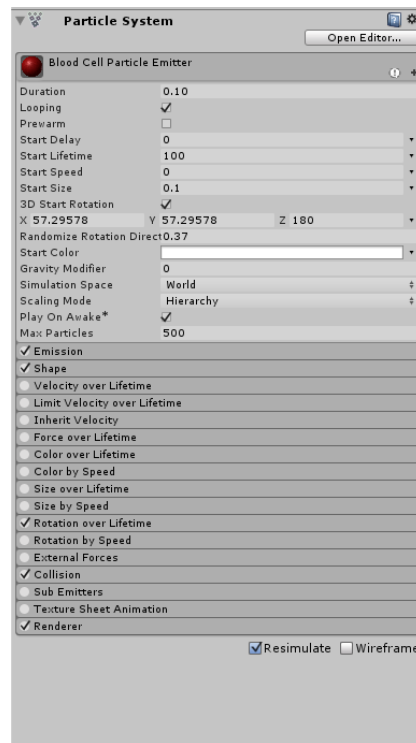


Abbildung 30: Partikelsystem-Komponente im Unity Editor.

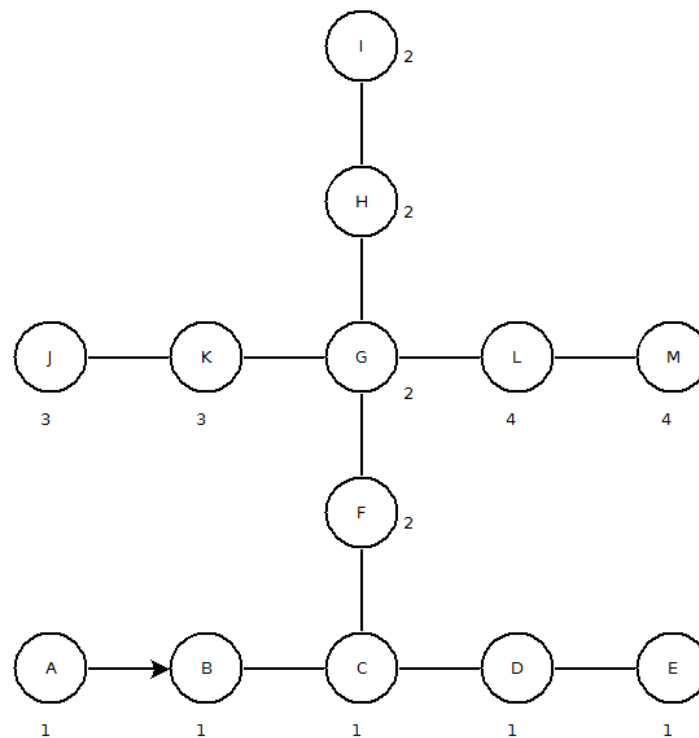
Das Hauptmodul enthält allgemeine Parameter über Partikel, wie Lebenszeit, Geschwindigkeit, Farbe und maximale Anzahl an Partikeln. Das Emission-Modul beeinflusst, wie viele neue Partikel in gewissen Zeitabständen erscheinen. Das Shape-Modul bestimmt die Form, aus der die Partikel emittiert werden. Das Collision-Modul ist für Partikelkollisionen zuständig und das Renderer-Modul bestimmt, wie ein einzelner Partikel dargestellt wird. Somit enthält das Shuriken Partikelsystem alles was nötig ist, um die Simulation zu implementieren, was Zeit spart.

5.1.2 Partikelpfadsystem

Das Prinzip der Simulation ist, eine Liste aller möglichen Pfade zu erstellen, die ein Partikel durch die Gefäßstruktur nehmen kann. Die Einträge dieser Liste sind Listen aus Richtungsvektoren zwischen zwei benachbarten Kontrollpunkten. Diese werden als Geschwindigkeitsvektor dem Partikel übergeben.

Zuerst muss bestimmt werden, in welcher Reihenfolge die Kontrollpunkte durchlaufen werden sollen. Hierfür wird eine Depth-First Traversierung durch den Graphen ausgehend vom Startpunkt und Richtung der platzierten Blutflussquelle durchgeführt. Der Vorteil an dieser Traversierung ist, dass zuerst alle Knoten einer Verzweigung durchlaufen werden, bevor die der anderen Verzweigung betrachtet werden. Dadurch ergibt sich für jeden Pfad die Reihenfolge der Knoten.

Während des Vorgangs wird der Wert einer Zählervariable den Kontrollpunkten als Pfadnummer zugewiesen. Diese Variable wird nur dann inkrementiert, wenn die Kontrollpunkte der nächsten Verzweigung betrachtet werden. Der Wert entspricht am Ende der Anzahl aller möglichen Pfade. Als weiteres Endergebnis entsteht eine Liste mit allen Kontrollpunkten in einer bestimmten Reihenfolge (siehe Abbildung 31).

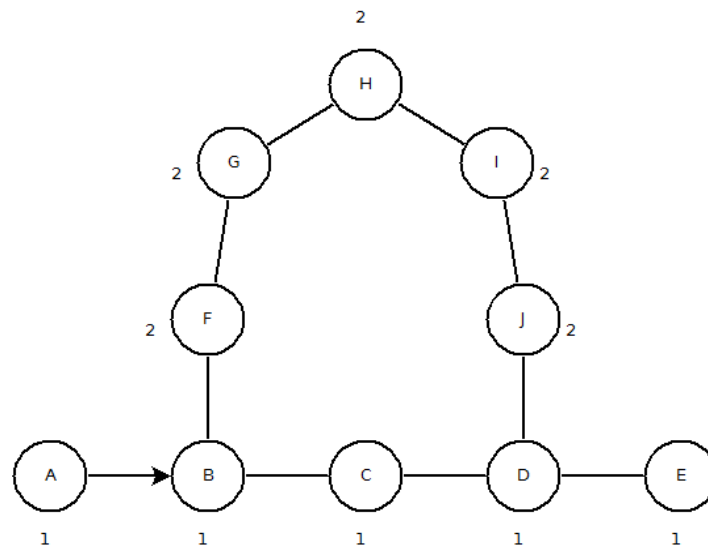


Result: A, B, C, D, E, F, G, H, I, K, J, L, M

Abbildung 31: Beispiel einer Depth-First Traversierung. Startpunkt und Richtung sind durch den Pfeil angegeben. Die Pfadnummern stehen unterhalb der Knoten.

Über diese Liste wird nun so oft iteriert wie es Pfade gibt. Für den in Abbildung 31 dargestellten Graphen wäre das viermal. In jeder Iteration wird eine neue Liste mit Richtungsvektoren erstellt und eine Hilfsvariable namens `_currentPathNumber` initialisiert. Diese Variable zeigt, welche Kontrollpunkte bei der Iteration betrachtet werden sollen. Wenn nun die Pfadnummer des aktuellen Kontrollpunktes gleich `_currentPathNumber` ist, dann wird die Anzahl der verbundenen Nachbarn überprüft. Ist diese kleiner oder gleich zwei, dann wird der Richtungsvektor zwischen dem aktuellen Kontrollpunkt und dem Nachbarn gebildet, der bei der aktuellen Iteration nicht bereits betrachtet wurde. Ist die Anzahl der Nachbarn größer als zwei, dann handelt es sich um eine Verzweigung. Nun muss entschieden werden, welcher Nachbar für die Berechnung des Richtungsvektors genommen werden soll. Hierfür werden alle Nachbarn betrachtet und überprüft, ob deren Pfadnummer kleiner oder gleich dem Index der aktuellen Iteration und größer oder gleich `_currentPathNumber` ist. Der Richtungsvektor wird mit dem Nachbarn gebildet, bei dem das zutrifft. Anschließend wird `_currentPathNumber` auf den Wert der Pfadnummer dieses Nachbarn gesetzt. Am Ende des gesamten Prozesses entsteht dann eine Liste mit allen möglichen Richtungsvektorlisten.

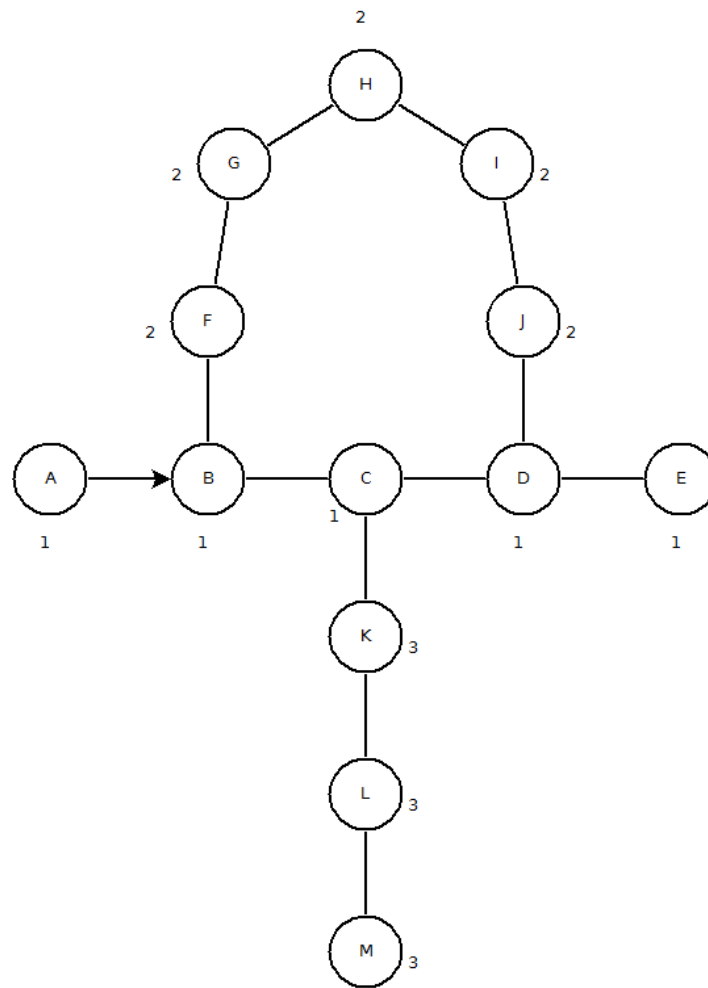
Bei Zyklen gibt es nun das Problem, dass bei der Liste der Depth-First Traversierung in solchen Fällen einige Einträge in der umgekehrten Reihenfolge liegen (siehe Abbildung 32). Dadurch würden sich die Partikel an der ersten Verzweigung aufteilen, aber dann in die falsche Richtung fließen und das Oberflächenmodell verlassen. Um dies zu unterbinden, wird für den aktuellen Kontrollpunkt überprüft, ob er den zuletzt betrachteten Kontrollpunkt als Nachbarn hat. Ist dies nicht der Fall, dann sind die folgenden Kontrollpunkte in der falschen Reihenfolge. Diese werden dann umgekehrt.



Result: A, B, C, D, E, J, I, H, G, F

Abbildung 32: Graph mit einem Zyklus. Nach der Depth-First Traversierung sind F, G, H, I, und J in der umgekehrten Reihenfolge.

Der Grund warum Pfade zwischen Zyklen ignoriert werden, liegt daran, dass `_currentPathNumber` nur inkrementiert werden kann (siehe Abbildung 33). Das bedeutet, dass ein Partikel niemals in einen Pfad mit kleinerer Pfadnummer als die aktuelle zurückkehren kann.



Result: A, B, C, D, E, J, I, H, G, F, K, L, M

Abbildung 33: Bei der dritten Iteration durch die Knoten um den dritten Pfad zu erreichen, wird zuerst der Knoten B betrachtet. Da F die Bedingung erfüllt, um mit B einen Richtungsvektor zu bilden, wird `_currentPathNumber` auf den Wert zwei gesetzt. Somit werden alle Knoten mit Pfadnummer drei ignoriert. Dadurch fließen die Partikel niemals durch diese Knoten durch.

5.1.3 Zuweisung der Pfade und Bewegen der Partikel

Alle Partikel des Shuriken Partikelsystems befinden sich in einem Array. Über dieses Array wird in jedem Frame iteriert und dabei wird jedem Partikel der Pfad zugewiesen, dem es folgen soll. Zur Bestimmung des korrekten Index der Pfadliste, wird zwischen dem aktuellen Array-Index und der Anzahl der Pfade eine Modulo-Rechnung durchgeführt. So wird sichergestellt, dass die Partikel nicht plötzlich zum falschen Pfad wechseln können.

Um nun zu bestimmen, welchen Richtungsvektor das Partikel nehmen soll, wird zuerst dessen Lebenszeit bestimmt. Als nächstes wird die zurückgelegte Distanz berechnet, indem die Summe der Gradienten der Richtungsvektoren gebildet wird. Die Lebenszeit bestimmt hierbei, bis zu welchem Index der Richtungsvektorliste aufsummiert wird. Anschließend wird eine Zählervariable initialisiert und die Distanz mit dieser Variable als Index mit der zurückgelegten Distanz verglichen

(siehe Abbildung 34). Die Zählervariable wird solange inkrementiert, bis dessen Distanz größer ist. Die Zählervariable ist dann der Index des Richtungsvektors, der dem Partikel als Geschwindigkeitsvektor zugewiesen wird. Somit bewegen sich die Partikel durch das Blutgefäß, bis sie den letzten Eintrag ihrer Pfadliste erreicht haben und danach verschwinden. Die Partikel brauchen jeweils eine Sekunde, um einen Richtungsvektor zu traversieren. Somit hängt die Geschwindigkeit von den Abständen zwischen den Kontrollpunkten ab.

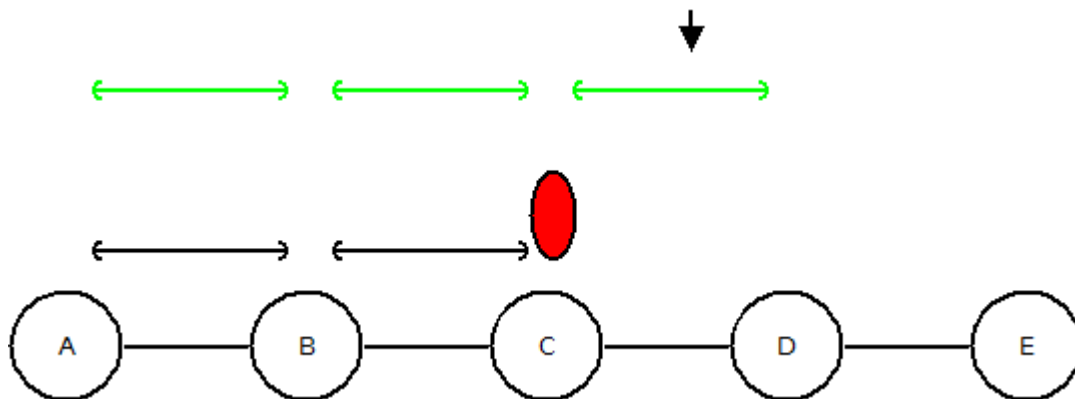


Abbildung 34: Der zurückgelegte Distanz des Partikels (hier in rot) ist durch die schwarzen Pfeile dargestellt. Die Distanz mit der Zählervariable als Index sind die grünen Pfeile. Sobald diese länger ist, als der Schwarze, wird dem Partikel den mit dem kleinen schwarzen Pfeil gekennzeichneten Richtungsvektor zugewiesen.

5.2 Kugelselektion

Die Implementierung ist dieselbe wie bei der alten Selektionsmethode. Es wurde lediglich das Selektionsvolumen ersetzt.

Die Aufhebung der gesamten Selektion erfolgt durch das Aktivieren eines Timers, wenn die entsprechende Taste gedrückt wird. Wenn dessen Wert beim Loslassen kleiner als eine Konstante ist, dann wird die Selektion aufgehoben. Das Problem hierbei kann sein, dass der Nutzer die Selektion aufhebt, wenn das nicht beabsichtigt wird. Dadurch müsste die Selektion nochmal von vorne begonnen werden, was umständlich ist. Alternativ könnte die Selektion aufgehoben werden, wenn die Taste gedrückt und losgelassen wird, während die Kugel mit keinem Kontrollpunkt kollidiert. Dies wurde jedoch nicht implementiert, da Unity über kein Attribut verfügt, mit der diese Bedingung überprüft werden kann.

Beim sofortigen Löschen und Änderung der Gefäßdicke wird lediglich selektiert und direkt im Anschluss die Aktion ausgeführt, wodurch der Prozess nahezu sofort passiert. Somit musste an der Implementierung keine Änderungen vorgenommen werden.

5.3 Verschieben der Blutgefäßstruktur

Mit GameObjects können Hierarchien aufgebaut werden, indem einem GameObject andere GameObjects als Kinder untergeordnet werden. Die Kinder bekommen dieselben Eigenschaften wie das Elternobjekt.

Die Gefäßstruktur ist solch ein Elternobjekt, das aus mehreren Liniensegmenten als Kindobjekten besteht. Diese Liniensegmente haben wiederum die Kontrollpunkte als Kindobjekte. Wenn nun ein Kontrollpunkt gegriffen wird, dann wird die Position und Rotation des Root-Objektes verändert. Dadurch würde sich eine Struktur, die nicht vom Stylus gegriffen wurde, ebenfalls bewegen. Zwischen den Liniensegmenten gibt es nämlich keine Hierarchien, da die Zuordnung bei Zyklen mehrdeutig sein kann. Würde der Nutzer nur einzelne Liniensegmente greifen, dann könnten Verbindungen gelöst werden.

6 Zusammenfassung und Ausblick

6.1 Zusammenfassung

In diesem Individualprojekt wurde der von Stojnic [6] entwickelte Prototyp um eine Partikelbasierte Blutflusssimulation und mit Hilfe von Patrick Saalefeld um Laden von Gefäßdatensätzen erweitert. Zusätzlich wurde die Kugelselektion als neue Selektionsmethode implementiert und ein Großteil der Mängel, die in der Evaluierung erwähnt worden waren, beseitigt.

Jedoch entspricht die Blutflusssimulation nicht dem realen Blutfluss im Körper. Zudem funktioniert die Simulation nicht fehlerfrei bei Zyklen. Beim Laden von Gefäßdatensätzen lassen sich die Gefäßmodelle nicht um neue Strukturen ergänzen und es gibt keine Möglichkeiten, Parameter innerhalb der Anwendung anzupassen.

Darüber hinaus gab es nicht mehr genügend Zeit, um eine erneute Evaluierung durchzuführen. Somit lassen sich keine Aussagen machen, ob die neuen Funktionen mit dem zSpace und den verwendeten Interaktionstechniken gut umgesetzt wurden und ob sie geeignet sind, die medizinische Ausbildung zu unterstützen. Zudem können keine Schlussfolgerungen gezogen werden, wie sehr sich die Verbesserungen auf die Leistungsfähigkeit des Nutzers auswirken.

6.2 Ausblick

Um die neuen Funktionen auf Benutzerfreundlichkeit und Eignung für Interaktionen in 3D UIs zu testen, sollte eine erneute Evaluierung durchgeführt werden. Da bei der Arbeit von Stojnic diese nur qualitativ war, wäre es angebracht, auch quantitative Methoden zu verwenden. Damit soll verglichen werden, wie sehr sich die Leistungsfähigkeit der Nutzer zwischen der alten und der neuen Version verbessert oder verschlechtert hat. Dabei sollten die Testpersonen aus Medizinstudierenden bestehen, da sie die Zielgruppe der Anwendung sind. Somit lässt sich zum einen herausfinden, ob die Anwendung zum Lernen von Gefäßstrukturen, Gefäßkrankheiten und Blutfluss geeignet ist. Zum anderen kann beobachtet werden, wie sehr Leute ohne Informatik-Hintergrund mit den Interaktionen des zSpace umgehen können.

Zur Verbesserung der Blutflusssimulation kann zum einen die Simulation realer gemacht werden, damit sie dem eigentlichen Blutfluss des Körpers entspricht. Dadurch könnten Medizinstudierende lernen, wie sich das Blut innerhalb des Gefäßes und den Gefäßwänden verhält und welche Auswirkungen Aneurysmen und Stenosen auf den Blutfluss haben. Davor sollte allerdings der aktuelle Algorithmus umgeschrieben werden, damit bei Zyklen keine Fehler mehr auftreten.

Das Platzieren von Blutflussquellen kann zudem erweitert werden, indem der Nutzer die Möglichkeit gegeben wird, z. B. durch Setzen von Pfeilen die Richtung des Blutflusses zu beeinflussen. Dadurch müssten Blutflussquellen nicht neu platziert werden und es wären Flusskorrekturen möglich.

Für das Laden von Datensätzen muss zunächst dafür gesorgt werden, dass die Liniensegmente, die entstehen, kontinuierlich sind, damit die Gefäße kompatibel für die Blutflusssimulation werden.

Zudem müssen eventuell Änderungen am virtuellen Stylus und am Zeichnen vorgenommen werden, damit die Gefäßmodelle um neue Strukturen ergänzt werden können. Weiterhin muss es bei der Benutzungsoberfläche Möglichkeiten zur Anpassung der Parameter geben. Hierfür wäre eventuell ein neuer Modus mit zusätzlichen Bedienelementen notwendig. Da die vorhandenen Elemente bereits viel Platz einnehmen, sollte darüber nachgedacht werden, ob diese nicht durch eine 2D-Benutzungsoberfläche ersetzt werden sollen, die sich an der Nullparallaxe befindet.

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] A. Pihuit, M.-P. Cani, und O. Palombi, „Sketch-based Modeling of Vascular Systems: A First Step Towards Interactive Teaching of Anatomy“, in *Proceedings of the Seventh Sketch-Based Interfaces and Modeling Symposium*, Aire-la-Ville, Switzerland, Switzerland, 2010, S. 151–158.
- [3] P. Clavert, J. Bouchaïb, F. Duparc, und J. L. Kahn, „A plea for the use of drawing in human anatomy teaching“, *Surg. Radiol. Anat.*, Bd. 34, Nr. 8, S. 787–789, Juni 2012.
- [4] P. Saalfeld, Stojnic, Aleksandar, B. Preim, und Oeltze, Steffen, „Semi-Immersive 3D Sketching of Vascular Structures for Medical Education“, in *Proceedings of Euroraphics Workshop on Visual Computing for Biology and Medicine*, Bergen, 2016.
- [5] M. Müller, S. Schirm, und M. Teschner, „Interactive blood simulation for virtual surgery based on smoothed particle hydrodynamics“, *Technol. Health Care*, Bd. 12, Nr. 1, S. 25–31, 2004.
- [6] Stojnic, Aleksandar, „Räumliches Skizzieren vaskulärer Strukturen und Pathologien mit dem 3D User Interface zSpace“, Bachelorarbeit, Otto-von-Guericke Universität, Magdeburg, 2016.
- [7] F. Argelaguet und C. Andujar, „A survey of 3D object selection techniques for virtual environments“, *Comput. Graph.*, Bd. 37, Nr. 3, S. 121–136, 2013.
- [8] B. Preim und R. Dachsel, *Interaktive Systeme*, 2., Bd. 2. Berlin, Heidelberg: Springer Berlin Heidelberg, 2015.
- [9] 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.
- [10] R. W. Lindeman, J. L. Sibert, und J. K. Hahn, „Hand-Held Windows: Towards Effective 2D Interaction in Immersive Virtual Environments“, in *Proceedings of the IEEE Virtual Reality*, Washington, DC, USA, 1999, S. 205–212.
- [11] D. E. Meyer, R. A. Abrams, S. Kornblum, C. E. Wright, und J. E. K. Smith, „Optimality in human motor performance: ideal control of rapid aimed movements“, *Psychol. Rev.*, Bd. 95, S. 340–370, 1988.
- [12] L. Vanacken, T. Grossman, und K. Coninx, „Exploring the effects of environment density and target visibility on object selection in 3D virtual environments“, in *2007 IEEE symposium on 3D user interfaces*, 2007.
- [13] T. Grossman und R. Balakrishnan, „The Design and Evaluation of Selection Techniques for 3D Volumetric Displays“, in *Proceedings of the 19th Annual ACM Symposium on User Interface Software and Technology*, New York, NY, USA, 2006, S. 3–12.

- [14] J. Liang und M. Green, „JDCAD: A highly interactive 3D modeling system“, *Comput. Graph.*, Bd. 18, Nr. 4, S. 499–506, Juli 1994.
- [15] G. de Haan, M. Koutek, und F. H. Post, „IntenSelect: Using Dynamic Object Rating for Assisting 3D Object Selection“, in *Proceedings of the 11th Eurographics Conference on Virtual Environments*, Aire-la-Ville, Switzerland, Switzerland, 2005, S. 201–209.
- [16] I. Poupyrev, S. Weghorst, M. Billinghurst, und T. Ichikawa, *Egocentric Object Manipulation in Virtual Environments: Empirical Evaluation of Interaction Techniques*. 1998.
- [17] M. R. Mine, „Virtual Environment Interaction Techniques“, University of North Carolina at Chapel Hill, Chapel Hill, NC, USA, 1995.
- [18] F. Steinicke, T. Ropinski, und K. Hinrichs, „Object selection in virtual environments using an improved virtual pointer metaphor“, in *Computer Vision and Graphics*, Springer, 2006, S. 320–326.
- [19] A. Steed und C. Parker, „3D Selection Strategies for Head Tracked and Non-Head Tracked Operation of Spatially Immersive Displays“, in *8th International Immersive Projection Technology Workshop*, 2004, S. 13–14.
- [20] R. Kopper, F. Bacim, und D. A. Bowman, „Rapid and accurate 3D selection by progressive refinement“, in *3D User Interfaces (3DUI), 2011 IEEE Symposium on*, 2011, S. 67–74.
- [21] C. Braley und A. Sandu, „Fluid Simulation For Computer Graphics: A Tutorial in Grid Based and Particle Based Methods“, *Va. Tech Blacksbg.*, 2010.
- [22] M. B. Liu und G. R. Liu, „Smoothed Particle Hydrodynamics (SPH): an Overview and Recent Developments“, *Arch. Comput. Methods Eng.*, Bd. 17, Nr. 1, S. 25–76, März 2010.
- [23] P. Saalfeld, „Methoden zur Skizzierung von Gefäßen und Gefäßerkrankungen mit integriertem Blutfluss“, Masterarbeit, Otto-von-Guericke Universität, Magdeburg, 2014.
- [24] J. Qin, W.-M. Pang, B. P. Nguyen, D. Ni, und C.-K. Chui, „Particle-based Simulation of Blood Flow and Vessel Wall Interactions in Virtual Surgery“, in *Proceedings of the 2010 Symposium on Information and Communication Technology*, New York, NY, USA, 2010, S. 128–133.
- [25] 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.
- [26] B. Preim und R. Dachsel, *Interaktive Systeme*, 2., Bd. 1. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010.
- [27] P. Shirley und S. Marschner, *Fundamentals of Computer Graphics*, 3rd Aufl. Natick, MA, USA: A. K. Peters, Ltd., 2009.
- [28] 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.

- [29] 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.
- [30] D. Hix und H. R. Hartson, *Developing User Interfaces: Ensuring Usability Through Product & Process*. New York, NY, USA: John Wiley & Sons, Inc., 1993.
- [31] K. Zilles, B. N. Tillmann, und Zilles-Tillmann, *Anatomie: mit 121 Tabellen*. Heidelberg: Springer, 2010.