

Otto-von-Guericke-Universität Magdeburg

Fakultät für Informatik

Institut für Simulation und Graphik



Masterarbeit

Vollautomatische Visualisierung von kardialen 4D PC-MRI Blutflussdaten mit intelligenter Fokussierung auf Features

Autor:

Benjamin Behrendt

17. April, 2015

Betreuer:

Prof. Dr.-Ing. Bernhard Preim

M. Sc. Benjamin Köhler

Gutachter:

Prof. Dr.-Ing. Bernhard Preim

Prof. Dr.-Ing. Holger Theisel

Behrendt, Benjamin:

Vollautomatische Visualisierung von kardialen 4D PC-MRI Blutflussdaten mit intelligenter Fokussierung auf Features

Masterarbeit, Otto-von-Guericke-Universität Magdeburg, 2015.

Inhaltsangabe

Verschiedene Pathologien im kardiovaskulären System können hämodynamische Veränderungen, wie die Entstehung von Verwirbelungen, auslösen. Mithilfe von 4D PC-MRI Bildgebung ist es möglich, den Blutfluss in Gefäßen zeitaufgelöst und nicht-invasiv zu akquirieren. Bisherige Ansätze zur Exploration dieser Daten erfordern vorwiegend eine Segmentierung oder verwenden direktes Volumenrendering in Verbindung mit Partikelsystemen, die keine automatische Hervorhebung interessanter Flusscharakteristika erlauben.

Einige medizinische Arbeiten korrelieren das Vorhandensein von Verwirbelungen mit verschiedenen Pathologien und führen zu diesem Zweck bisher nur qualitative Auswertungen von Blutflussdaten durch. Eine durch Segmentierung ermöglichte quantitative Analyse ist hierfür nicht erforderlich, die Hervorhebung von Features, wie Verwirbelungen oder Rückflüssen, hingegen hilfreich.

Das Ziel dieser Arbeit ist die Entwicklung und Evaluation einer Methode zur Visualisierung von kardialen 4D-PC-MRI-Daten. Diese soll durch Automatisierung, Echtzeitfähigkeit durch Verwendung von GPU-Unterstützung sowie Hervorhebung von Features des Blutflusses, wie Flussgeschwindigkeiten, Verwirbelungen oder Rückflüssen, eine effiziente Exploration sowie qualitative Analyse des Blutflusses ermöglichen.

Um die in dieser Arbeit vorgestellte Methode zu evaluieren, wurde ein Prototyp entwickelt, der von medizinisch nicht geschulten Probanden getestet wurde. Die Ergebnisse dieses Tests zeigen, dass unter Verwendung dieser Methode eine qualitative Untersuchung des Blutflusses ohne Segmentierung möglich ist, solange der verwendete Datensatz einen ausreichenden Kontrast aufweist. In 73.6% der Fälle waren die Probanden in der Lage, die Verwirbelungen in einem Datensatz vollständig zu erkennen, in nur 11.1% der Untersuchungen wurden Flussstrukturen fälschlicherweise als Verwirbelungen erkannt.

Inhaltsverzeichnis

Abbildungsverzeichnis	ix
Tabellenverzeichnis	x
1 Einführung	1
2 Grundlagen	3
2.1 Medizinische Grundlagen	3
2.1.1 Blutkreislauf	3
2.1.2 Kardiovaskuläre Erkrankungen	4
2.1.3 4D PC-MRI	6
2.2 Mathematische Grundlagen	7
2.2.1 Integration in Vektorfeldern	7
2.2.2 Jacobi-Matrix	9
2.3 Technische Grundlagen	10
2.3.1 Shader-Modell	10
2.3.1.1 Rendering-Pipeline	10
2.3.1.2 Multipass-Rendering	12
2.3.1.3 General Purpose GPU	12
2.3.2 Indirect Volume Rendering	13
2.3.3 Direct Volume Rendering	14
2.3.4 Transferfunktionen	18
3 Verwandte Arbeiten	21
3.1 Klinische Studien mittels 4D PC-MRI	21
3.2 Visualisierung von Flussdaten	23
3.3 Wirbelextraktion	26
4 Vollautomatische Visualisierung von kardialen 4D PC-MRI Blut- flussdaten mit intelligenter Fokussierung auf Features	31
4.1 DVR	31
4.1.1 Rendering-Pipeline	32
4.1.1.1 Clearing und Farbwürfel	32
4.1.1.2 Volume Rendering	33
4.1.2 Darstellung der Anatomie	34
4.1.2.1 MIP	34
4.1.2.2 Erweiterte Rampenfunktion	34
4.1.2.3 Solider Körper	36

4.1.3	Automatische Transferfunktionen und Schwellenwerte	36
4.1.4	Tiefenberechnung	38
4.2	Partikelvisualisierung	38
4.2.1	Seeding der Partikel	39
4.2.2	Berechnung der Partikelpfade	40
4.2.3	Visualisierung der Partikel	41
4.3	Komposition von Partikeln und DVR	45
4.3.1	Komposition anhand der Tiefenwerte	45
4.3.2	DVR-Frontface-Culling	46
4.4	Vorberechnungen	48
4.4.1	Verbesserung der Flussbilder	48
4.4.2	Flussrichtung	48
4.4.3	Wirbelkriterium	49
5	Ergebnisse und Diskussion	50
5.1	Datensätze	50
5.2	Visualisierung der Anatomie	51
5.3	Untersuchung der Datensätze	55
5.3.1	Versuchsaufbau	55
5.3.2	Dauer der Untersuchungen	56
5.3.3	Ergebnisse der Wirbelfindung	57
5.3.4	Befragungsergebnisse	69
5.4	Performance	72
6	Zusammenfassung und Ausblick	75
	Literaturverzeichnis	77

Abbildungsverzeichnis

2.1	Schematische Darstellung des menschlichen Kreislaufsystems mit Körper- und Lungenkreislauf (links), Aufbau des menschlichen Herzens (rechts).	4
2.2	Ermittlung der Einflüsse der Eddy Currents durch Analyse des statischen Gewebes. [HFS ⁺ 11]	6
2.3	Detektion und Entfernung von Phase Wraps. [HFS ⁺ 11]	7
2.4	Trilineare Interpolation. 8 Gitterpunkte (links) werden verwendet, um über mehrere lineare Interpolationsschritte (rechts) den Wert eines beliebigen Punktes p_{query} zu bestimmen. [Köh12]	8
2.5	Euler-Integration: Ein neuer Punkt wird aus dem alten Punkt und dessen Vektor bestimmt. [Köh12]	8
2.6	Visualisierung des Funktionsprinzips von RK2 (links) sowie RK4 (rechts). [Köh12]	9
2.7	Shader-Pipeline zur Umwandlung von Grafikprimitiven (links) in Pixel (rechts). [Rák10]	11
2.8	Vergleich der Beleuchtungseffekte zwischen <i>Gouraud Shading</i> (links) und <i>Phong Shading</i> (rechts). [Sch10]	11
2.9	Tiefenwerte einer 3D-Szene im Z-Buffer. Nahe Objekte erscheinen hell, weiter entfernte dunkel	12
2.10	Texturbasiertes DVR; Die Visualisierung wird durch das Überlagern einzelner Texturschichten erzeugt, die entlang der Blickrichtung (<i>view-aligned</i> , oben) oder der Koordinatenachsen (<i>axis-aligned</i> , unten) ausgerichtet sind [EE02].	15
2.11	Darstellung der Bestimmung der Koordinatenachsen für die Berechnung des Strahls [Gro10].	15
2.12	Darstellung der Bestimmung des Ursprungspunktes der Bildebene sowie Iteration über alle Pixel [Gro10].	16
2.13	Rendering des Farbwürfels: Backface-Culling (links) und Frontface-Culling (rechts) [KW03]	17

2.14	Grafischer Transferfunktionen-Editor aus MeVisLab mit verschiedenen eingestellten Transferfunktionen. Die Visualisierung zeigt Aorta und Pulmonalarterie.	18
2.15	Bestimmung einer Transferfunktion durch automatische Generierung verschiedener Funktionen und Auswahl der besten Funktion durch den Nutzer. [KG01]	19
2.16	Zweidimensionale Transferfunktionen auf einem ebenfalls zweidimensionalen Histogramm sowie deren Auswirkungen auf die Visualisierung [SKGM03].	19
2.17	Verschiedene Störeinflüsse auf zweidimensionalen Histogrammen, die Intensität und Gradienten darstellen [SVSG06].	20
3.1	Funktionsweise des Konvolutionsverfahrens nach Van Wijk (a), Anwendung des Verfahrens auf einen circulären und einen turbulenten Fluss (b), Anwendung des verfeinerten Verfahrens von Cabral und Leedom (c). Quelle: [CL93]	23
3.2	Mehrfarbige Illuminated Streamlines mit Halos [MTHG03].	25
3.3	Visualisierung von Wirbelkernen alleinstehend oder zusammen mit Pfadlinien nach Stalder et al. [SFH ⁺ 10].	26
3.4	Clustering eines Vektorfeldes (a) sowie Repräsentation jedes Clusters durch einen einzelnen Vektor (b). Quelle: [HWHJ99]	26
3.5	Falscherkennung von Wirbeln in einem Rohr mit laminarem Fluss; Parabolisches Geschwindigkeitsprofil (links) und Ebene mit Geschwindigkeiten (Mitte), Stärke der Vorticity (rechts)) [Kha11]	27
3.6	Fehlerhaft erkannter Wirbelkern (rot) in helikalem Fluss [Rot00].	28
3.7	Schrittweise Verfolgung der Wirbelkernlinie mittels <i>Prediction-Corrector</i> [BS95].	29
3.8	Visualisierung einer Aorta mit enthaltenen Wirbeln nach Köhler [Köh12].	30
4.1	Einfluss der verschiedenen vordefinierten Schrittweiten auf unterschiedliche Visualisierungsformen.	33
4.2	Schematische Darstellung einer üblichen Rampenfunktion (a), sowie der verwendeten erweiterten Rampenfunktion mit Opazität (rot) und Grauwert (blau) (b), beide gekennzeichnet durch den Beginn der Rampe (b_1) sowie deren Ende (b_2).	35
4.3	Vergleich unterschiedlicher Exponenten für den Grauwert-Teil der Rampenfunktion.	36
4.4	Vergleich der verschiedenen Nachbearbeitungsschritte bei der soliden Visualisierung: (a) Ohne Nachbearbeitung, (b) mit Rändern sowie (c) mit Rändern und Beleuchtung	37

4.5	Histogramm eines Datensatzes (gelb) mit eingezeichneten Quantilen für Rampenfunktion (orange) sowie solide Körper (grün).	37
4.6	Tiefenwert-Artefakte. Obwohl beide Punkte (rot) auf der Anatomie (orange) gleich weit vom Betrachter entfernt sind, entstehen aufgrund der Ausrichtung des Volumens (hellgrün) unterschiedliche Tiefenwerte (blau).	38
4.7	Ausfüllung des Gefäßes mit Pfadlinien bei unterschiedlicher Partikelzahl	39
4.8	Ungleichmäßiges Seeding: Gleichmäßiger Seeding-Bereich (weiß), Bereiche mit hohem λ_2 -Wert (rot), zusätzliche Seeding-Maske (grün). . .	40
4.9	Vergleich von vorberechneten mit in Echtzeit berechneten λ_2 -Werten. Zu laminarem Fluss gehörende Linien sind ausgeblendet, der Fluss in diesen Bereichen wird stattdessen durch gelbe Richtungspfeile visualisiert.	42
4.10	Vergleich der verschiedenen Nachbearbeitungsschritte beim Zeichnen der Pfadlinien.	43
4.11	Partikelvisualisierung (a) mit und (b) ohne Overlay. Innenliegende Teile des Wirbels sind ohne Overlay schlechter erkennbar.	44
4.12	Entfernung einer unterschiedlichen Anzahl von Pfadlinien in laminarem Fluss.	44
4.13	(a) Visualisierung des Gefäßes ohne Partikel und Frontface-Culling; (b) durch Gefäßanatomie verdeckte Partikel; (c) Rückwand des Gefäßes; (d) durch Frontface-Culling sichtbare Partikel.	47
4.14	Frontface-Culling; Aktueller Punkt (blau) mit Analysebereich (rot). Alle Voxel vor dem aktuellen Punkt werden mit reduzierter Opazität gezeichnet (grüner Strahl). Da sich die letzten 10% des Analysebereichs unterhalb des Schwellenwertes (grauer Kreis) befinden, werden alle nachfolgenden Voxel mit voller Opazität gezeichnet.	48
5.1	Visualisierung des Datensatzes <i>Aortenbypass_1_2011</i> in den drei Varianten MIP (a), Rampenfunktion (b) und solider Körper (c)	52
5.2	Visualisierung des Datensatzes <i>2_2012_Fallot</i> in den drei Varianten MIP (a), Rampenfunktion (b) und solider Körper (c). Deutlich erkennbar sind Teile des rechten Ventrikels unten links im Bild.	52
5.3	Visualisierung des Datensatzes <i>Flow_MD_1_2014</i> in den drei Varianten MIP (a), Rampenfunktion (b) und solider Körper (c).	53
5.4	Visualisierung des Datensatzes <i>Flow_MD_3_2014</i> in den drei Varianten MIP (a), Rampenfunktion (b) und solider Körper (c). Visualisierung des Datensatzes mit manueller Transferfunktion (d). Deutlich erkennbar sind die Artefakte im linken Teil der Bilder sowie der fehlende Aortenbogen in (b) und (c).	53

5.5	Visualisierung des Datensatzes <i>ProtheseISTA_11_2012</i> in den drei Varianten MIP (a), Rampenfunktion (b) und solider Körper (c). Visualisierung des Datensatzes mit manueller Transferfunktion (d).	54
5.6	Visualisierung des Datensatzes <i>Flow_MD_4_2014</i> in den drei Varianten MIP (a), Rampenfunktion (b) und solider Körper (c).	54
5.7	Durchschnittliche Bearbeitungsdauer in Minuten pro Datensatz, geordnet nach Zeit.	57
5.8	Durchschnittliche Bearbeitungsdauer in Minuten der Teilnehmer (P1 - P8).	58
5.9	Durchschnittliche Bearbeitungsdauer in Minuten pro Darstellungsart (a), Konfiguration (b) sowie Datensatzgruppe (c).	58
5.10	Skizzen der Teilnehmer zu den Probanden-Datensätzen <i>anonymous</i> (oben), <i>Aorta_13_2012</i> (Mitte) sowie <i>Flow_MD_1_2014</i> (unten). . . .	59
5.11	<i>2_2012_Fallot</i> mit Darstellung der Verwirbelungen ohne laminaren Fluss (a), mit vorberechneten λ_2 -Werten und Overlays (b), in Echtzeit berechneten λ_2 -Werten ohne Overlays (c) sowie den Wirbelskizzen der Teilnehmer (unten).	60
5.12	<i>Aortenbypass_1_2011</i> mit Darstellung der Verwirbelungen ohne laminaren Fluss (a), mit vorberechneten λ_2 -Werten (b), in Echtzeit berechneten λ_2 -Werten (c) sowie den Wirbelskizzen der Teilnehmer (unten).	61
5.13	<i>Aortenektasie_04_2011</i> mit Darstellung der Verwirbelungen ohne laminaren Fluss (a), mit vorberechneten λ_2 -Werten (b), in Echtzeit berechneten λ_2 -Werten (c) sowie den Wirbelskizzen der Teilnehmer (unten).	62
5.14	<i>Aortenektasie_09_2012</i> mit Darstellung der Verwirbelungen ohne laminaren Fluss (a), mit vorberechneten λ_2 -Werten (b), in Echtzeit berechneten λ_2 -Werten (c) sowie den Wirbelskizzen der Teilnehmer (unten).	63
5.15	<i>Flow_MD_2_2014</i> mit Darstellung der Verwirbelungen ohne laminaren Fluss (oben) sowie den Wirbelskizzen der Teilnehmer (unten). . .	64
5.16	<i>Flow_MD_4_2014</i> mit Darstellung der Verwirbelungen ohne laminaren Fluss (a), mit vorberechneten λ_2 -Werten (b), in Echtzeit berechneten λ_2 -Werten (c) sowie den Wirbelskizzen der Teilnehmer (unten). . .	65
5.17	<i>ISTA_06_2012</i> mit Darstellung der Verwirbelungen ohne laminaren Fluss (a), mit vorberechneten λ_2 -Werten (b), in Echtzeit berechneten λ_2 -Werten (c) sowie den Wirbelskizzen der Teilnehmer (unten). . . .	66
5.18	<i>ISTA_08_2012</i> mit Darstellung der Verwirbelungen ohne laminaren Fluss (a), mit vorberechneten λ_2 -Werten (b), in Echtzeit berechneten λ_2 -Werten (c) sowie den Wirbelskizzen der Teilnehmer (unten). . . .	67

5.19	<i>ProtheseISTA_11_2012</i> mit Darstellung der Verwirbelungen ohne laminaren Fluss (oben) sowie den Wirbelskizzen der Teilnehmer (unten).	68
5.20	Verdeckungen im Datensatz <i>2_2012_Fallot</i> (a). Ohne Overlays sind die Verdeckungsverhältnisse leicht erkennbar (b), bei aktivierten Overlays ist die Zuordnung der Wirbel zu Aorta oder Pulmonalarterie wesentlich schwieriger (c).	70
5.21	Allgemeine Bewertung des Prototypen sowie der Visualisierung auf einer Skala von 0 (negativ) bis 4 (positiv).	71
5.22	Bewertung der einzelnen Darstellungsarten auf einer Skala von 0 bis 4.	71
5.23	Durchschnittliche Bewertung der Darstellungsarten pro Anwender auf einer Skala von 0 bis 4.	72
5.24	Unterschiedliche Blickwinkel und Skalierungen und ihre Auswirkungen auf die Framerate.	74

Tabellenverzeichnis

5.1	Übersicht aller verwendeten Datensätze.	50
5.2	FPS für unterschiedliche Datensätze und Darstellungsarten bei aktivierter Flussvisualisierung mit 4096 Partikeln. Für solide Körper und Rampenfunktion ist zusätzlich die FPS-Zahl bei aktiviertem Frontface-Culling angegeben. Die Auflösung des Rendering-Bereichs betrug 828 x 918 Pixel.	73
5.3	FPS für unterschiedliche Datensätze und Darstellungsarten ohne Partikelvisualisierung. Die Auflösung des Rendering-Bereichs betrug 828 x 918 Pixel.	73

1. Einführung

Blut ist ein zentraler Bestandteil des Körpers, der eine Vielzahl lebenswichtiger Funktionen erfüllt. Unter anderem dient es dem Transport von Nährstoffen, Sauerstoff und Enzymen sowie dem Abtransport von Stoffwechselprodukten wie Kohlenstoffdioxid oder Harnstoff.

Um sämtliche Zellen des Körpers erreichen zu können, wird das Blut durch ein komplexes Kreislaufsystem geleitet, in dessen Zentrum das Herz liegt. Dieses schlägt etwa 70 Mal pro Minute und pumpt dabei jedes Mal zwischen 70 und 100 ml Blut durch den Körper.

Störungen des Herzens oder der Blutgefäße können weitreichende Konsequenzen nach sich ziehen. So führt beispielsweise ein Gefäßverschluss zu einer Sauerstoff-Unterversorgung, durch welche sich nekrotisches Gewebe bilden kann. Pathologien der Gefäßwand beeinträchtigen dessen Stabilität, wodurch das Risiko einer Ruptur steigt. Eine Dissektion der Aorta endet zumeist tödlich. Aus diesem Grund ist es wichtig, Abnormalitäten in Gefäßen möglichst frühzeitig zu erkennen, um weitere Schädigungen zu vermeiden.

Klassische Methoden zur Diagnose solcher Pathologien stützen sich auf *Doppler-Sonografie* oder den Einsatz von Kathetern und Kontrastmittel. Ersteres erlaubt jedoch nur die Erfassung von Blutströmungen in der selben Richtung wie die Schallwellen [DKS⁺08], während letzteres einen invasiven Eingriff darstellt.

Phase Contrast Magnetic Resonance Imaging (kurz PC-MRI) ermöglicht es hingegen, den Blutfluss eines Patienten nichtinvasiv zu erfassen. Dabei werden Phasenunterschiede zwischen dem in Bewegung befindlichen Blut und dem umgebenden, stationären Gewebe ausgenutzt, um den Blutfluss zu erfassen. Werden mehrere Aufnahmen zu unterschiedlichen Zeiten gemacht, so spricht man von *2D* oder *4D PC-MRI*. Während bei 2D PC-MRI lediglich eine einzelne vor der Aufnahme konfigurierte Sicht akquiriert wird, kann mittels 4D PC-MRI ein 3D-Volumen untersucht werden.

Bisherige Ansätze zur Visualisierung und quantitativen Auswertung von 4D PC-MRI-Daten, wie beispielsweise von [Köh12], erfordern ein manuelles oder semi-

automatisches Segmentieren der Gefäße. Je nach Qualität der Daten und Art der Segmentierung kann dies eine zeitaufwendige Arbeit sein. Verfahren zur rein qualitativen Auswertung, wie von [SGGJ13], erfordern zwar keine Segmentierung, erlauben dafür jedoch keine oder nur sehr eingeschränkte Hervorhebung von Features.

Das Ziel dieser Arbeit ist es, eine Methode zur Visualisierung von Flussdaten zu entwickeln, die ohne vorhergehende Segmentierung eine Exploration der Daten erlaubt und diese durch intelligente Hervorhebung von Features, wie Flussgeschwindigkeiten oder Verwirbelungen, unterstützt. Die qualitative Untersuchung von Flussdaten ohne die Durchführung einer Segmentierung spart Zeit bei der Auswertung medizinischer Studien, die Verwirbelungen im Blutfluss mit bestimmten Pathologien korrelieren.

Visualisierungsparameter sollen automatisch so festgelegt werden, dass sowohl die Anatomie als auch der Fluss in hoher Qualität dargestellt werden. Dies geschieht mithilfe einer Histogrammanalyse der Bildintensitätswerte. Dennoch kann der Anwender die Darstellung auch manuell anpassen, um Defizite der automatischen Erkennung auszugleichen.

Kapitel 2 bietet einen Überblick über medizinische, mathematische und technische Hintergründe. Ein Fokus liegt hierbei auf der Anatomie des kardiovaskulären Systems sowie mögliche Erkrankungen und deren Auswirkungen. Zudem werden die bei 4D PC-MRI-Untersuchungen anfallenden Daten sowie für deren Visualisierung verwendete Funktionen der Grafikhardware beschrieben.

Anschließend folgt ein Überblick über relevante Arbeiten im Bereich der Analyse sowie Visualisierung von Blutflussdaten. Dieser enthält sowohl Arbeiten über die hier verwendete Partikelintegration als auch alternative Methoden zur Visualisierung von Flussdaten. Zudem werden Arbeiten über die automatische Extraktion von Verwirbelungen behandelt. Auch auf medizinische Studien, welche bestimmte Pathologien des kardiovaskulären Systems mit spezifischen Blutflussmuster korrelieren, wird in diesem Kapitel eingegangen.

Im Rahmen dieser Arbeit wurde ein entsprechender Prototyp entwickelt, der die Analyse von 4D PC-MRI-Daten sowie Hervorhebung von Features ohne vorhergehende Segmentierung ermöglicht. Die für diesen Prototypen verwendeten Techniken und Algorithmen werden in Kapitel 4 vorgestellt. Neben Standardtechniken wie direktem Volumenrendering werden hier auch eigens für diesen Prototypen entwickelte Verfahren diskutiert, wie beispielsweise die Anwendung von Frontface-Culling bei direktem Volumenrendering.

Die Ergebnisse der Anwendung des Prototypen auf Patientendatensätze werden in Kapitel 5 vorgestellt. Dies umfasst sowohl die Leistungen als auch Beschränkungen des Prototypen im Bezug auf die Darstellung der Anatomie sowie des Blutflusses. Ebenfalls wird hier eine Probandenstudie präsentiert, bei der acht Personen mithilfe des Prototypen Wirbelstrukturen in Datensätzen unter Verwendung verschiedener Visualisierungsparameter erkennen mussten. Dabei zeigte sich, dass Verwirbelungen im visualisierten Blutfluss auch für Probanden ohne medizinische Ausbildung gut erkennbar sind. Eine Bedingung dafür ist jedoch ein ausreichender Kontrast in den Datensätzen.

2. Grundlagen

Der folgende Abschnitt beinhaltet einen Überblick über verschiedene medizinische, mathematische und technische Hintergründe dieser Arbeit. Zunächst werden die Anatomie des kardiovaskulären Systems sowie mögliche Erkrankungen und deren Auswirkungen beschrieben. Weiterhin wird auf 4D PC-MRI-Untersuchungen und die Verarbeitung der dabei anfallenden Daten eingegangen. Abschließend werden die technischen Grundlagen für die Visualisierung dieser Daten mithilfe der Grafikkarte vorgestellt.

2.1 Medizinische Grundlagen

2.1.1 Blutkreislauf

Der Blutkreislauf des Menschen besteht aus dem Herzen sowie einer Vielzahl von Blutgefäßen, die sich durch den gesamten Körper ziehen. Diese formen den sauerstoffreichen Körperkreislauf sowie den sauerstoffarmen Lungenkreislauf. [Abbildung 2.1](#) zeigt auf der linken Seite eine schematische Darstellung beider Kreisläufe.

Abhängig von seinem Körpergewicht verfügt ein erwachsener Mensch über ca. fünf bis sechs Liter Blut. Neben Nährstoffen, Botenstoffen und Stoffwechselendprodukten transportiert es den über die Lunge aufgenommenen Sauerstoff durch den Körper, wo er von den Zellen aufgenommen und verarbeitet wird. Das dabei entstehende Kohlenstoffdioxid wird daraufhin zurück zur Lunge getragen, wo es aus dem Körper ausgestoßen wird. Zusätzlich enthält das Blut Immunzellen zur Abwehr von Fremdkörpern wie Viren und verschließt Verletzungen durch Gerinnung.

Das menschliche Herz enthält zwei Kammern (linker und rechter *Ventrikel*) mit jeweils einem dazugehörigen Vorhof. Durch die Herzscheidewand sind beide Bereiche vollständig voneinander getrennt, sodass sich sauerstoffreiches und sauerstoffarmes Blut nicht vermischen können. Eine schematische Darstellung des Herzens ist rechts in [Abbildung 2.1](#) dargestellt.

Ziehen sich die Kammern während des Herzschlages zusammen (*Systole*), wird sauerstoffreiches Blut in die Aorta und sauerstoffarmes in die Pulmonalarterie gepumpt.

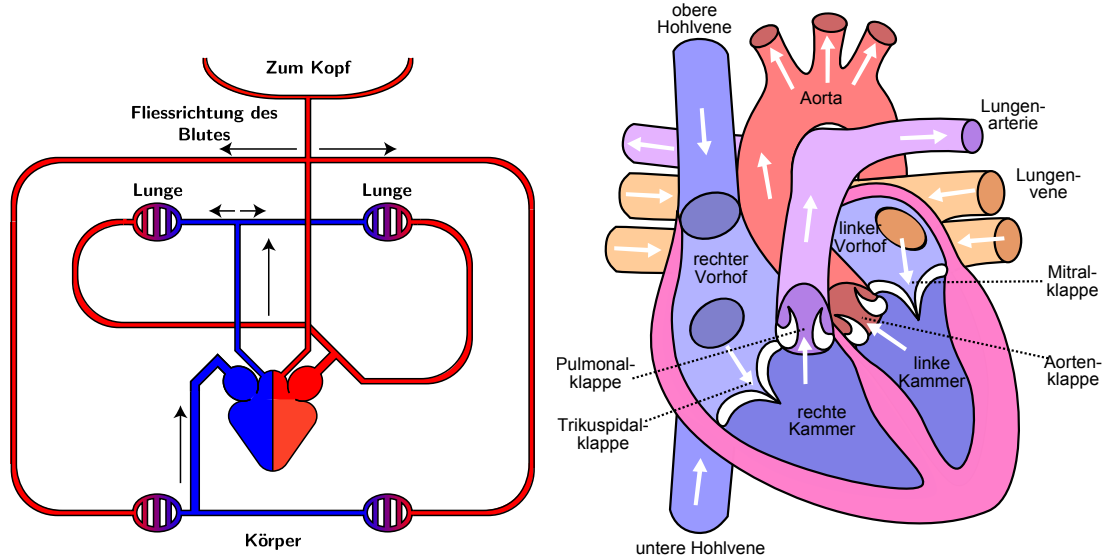


Abbildung 2.1: Schematische Darstellung des menschlichen Kreislaufsystems mit Körper- und Lungenkreislauf (links, Quelle: Wikipedia¹), Aufbau des menschlichen Herzens (rechts, Quelle: Wikipedia²).

Während dieser Phase entspannen sich die Vorhöfe, sodass über die Venen neues Blut in sie einströmen kann. In der nächsten Phase des Herzschlags (*Diastole*) entspannen sich die Herzkammern, während sich die Vorhöfe zusammenziehen. So kann das Blut von den Vorhöfen in die Kammern fließen, um das Herz von dort aus während der nächsten Systole zu verlassen. Um zu verhindern, dass das Blut von den Kammern in die Vorhöfe strömt, sind diese durch Herzklappen voneinander getrennt, die das Blut nur in eine Richtung passieren lassen.

2.1.2 Kardiovaskuläre Erkrankungen

Häufig resultieren Erkrankungen der Gefäße in einer Veränderung des Gefäßdurchmessers. Vergrößert sich dieser, beispielsweise aufgrund einer Schwächung oder Schädigung der Gefäßwand [KGP⁺13], so spricht man von einer *Ektasie*, ab einer Vergrößerung um mehr als den Faktor 1.5 von einem *Aneurysma*. Auch Bluthochdruck kann zu Weitungen des Gefäßes führen. Begünstigt wird die Bildung von Aneurysmen zusätzlich durch das *Marfan-Syndrom*, eine genetisch vererbte Bindegewebsschwäche, welche neben Augen und Skelettsystem auch kardiovaskuläre Blutgefäße beeinflusst [Die14].

Die Bildung von Aneurysmen kann zu weiteren Komplikationen führen. So besteht beispielsweise das Risiko einer *Aortendissektion*, bei der die innere Gefäßwand (*Intima*) einreißt und es zu Wandeinblutungen kommt [WNR⁺08]. Diese wiederum können zu einer Ruptur der Aorta und damit in den meisten Fällen zum Tod führen.

¹https://upload.wikimedia.org/wikipedia/commons/0/0c/Blutkreislauf_Gleichwarmer.svg (Gemeinfrei)

²https://upload.wikimedia.org/wikipedia/commons/3/33/Diagram_of_the_human_heart_%28cropped%29_de.svg (CC BY-SA 3.0, Autor: Jakob)

Bei einer krankhaften Verringerung des Gefäßdurchmessers hingegen spricht man von einer *Stenose*. Diese kann unter Anderem als Folge einer *Arteriosklerose* auftreten. Hierbei lagern sich sog. *Plaques*, die aus verschiedenen Stoffen wie Cholesterin oder Fettsäuren bestehen, an der Arterienwand ab. Dies blockiert den Blutfluss und führt damit zu Verwirbelungen [BBV04] sowie einer Steigerung der Flussgeschwindigkeit und des Drucks, da dieselbe Menge an Blut nun durch ein kleineres Gefäß gepumpt werden muss. Bei zunehmender Blockade oder einem vollständigen Verschluss des Gefäßes kommt es zu Versorgungsengpässen von Gewebe. Ein Spezialfall der Stenose ist die Aortenisthmusstenose (*ISTA*), bei der die Verengung im Aortenbogen auftritt.

Eine Stenose kann nicht nur im Gefäß selbst, sondern auch in dessen Klappe auftreten. In diesem Fall spricht man von einer Aorten- oder Pulmonalklappenstenose. Da sich derartige Stenosen für gewöhnlich nur sehr langsam entwickeln, kann das Herz die verringerte Öffnungsfläche durch höheren Pumpdruck kompensieren. Die Stenose bleibt in diesem Fall mitunter Jahre oder sogar jahrzehntelang asymptomatisch. Das gestörte Klappenöffnungsverhalten kann auch hier zu Verwirbelungen führen [CLT06].

Schließen die Gefäßklappen nicht mehr vollständig, so spricht man von Aorten- bzw. Pulmonalklappeninsuffizienz. In einem solchen Fall fließt während der Diastole Blut aus der Aorta bzw. Pulmonalarterie in die Herzkammern zurück. Dadurch steigt die Volumenbelastung der Herzkammer, die darauf mit einer Vergrößerung (*Hypertrophie*) reagiert. Als Folge daraus kann sich durch die Überbelastung des Herzmuskels eine Herzinsuffizienz herausbilden. Hier ist das Herz nicht mehr in der Lage, genügend Blut zur Versorgung des Körpers durch die Gefäße zu pumpen.

Bei *bikuspiden Gefäßklappen* (*BAV*) sind zwei der drei Semilunarklappen, aus denen die Gefäßklappe besteht, miteinander verwachsen. Bei einem Großteil der Betroffenen führt dies zu einem veränderten Blutflussbild mit Verwirbelungen [BHB⁺13]. Bikuspidale Klappen können sowohl angeboren als auch erworben werden (funktionelle BAV) und treten bei 0.5 bis 1.5% der Bevölkerung auf [BMB⁺12]. Sie stehen im Verdacht, die Entstehung anderer Erkrankungen im kardiovaskulären System zu begünstigen. Dazu zählen Klappenstenosen, Aneurysmenbildung sowie Gefäßdissektionen [CLT06] [BMB⁺12].

Ein komplexer kardiovaskulärer Geburtsdefekt ist die *Fallot-Tetralogie*, die nach dem französischen Pathologen Etienne-Louis Fallot benannt ist [AWR09]. Sie ist durch vier verschiedene Pathologien gekennzeichnet. Das erste ist eine Pulmonalstenose, durch die der Blutfluss in die Pulmonalarterie erschwert wird. Zudem tritt ein Ventrikelseptumdefekt auf, wodurch beide Herzkammern nicht mehr voneinander getrennt sind. Über diesem Defekt bildet sich eine reitende Aorta aus, was zu einer Vermischung von sauerstoffreichem und sauerstoffarmen Blut in der Aorta führt. Dieser Effekt wird durch die Blockade der Pulmonalarterie zusätzlich verstärkt, es bildet sich demzufolge ein Rechts-Links-Shunt heraus [BA09]. Als Folge dieser Veränderungen tritt zudem eine Hypertrophie des rechten Ventrikels auf. Üblicherweise werden Ventrikelseptumdefekt und Pulmonalstenose kurz nach der Geburt operativ korrigiert, allerdings führt dieser Eingriff im späteren Leben häufig zum Auftreten von Rückflüssen in der Pulmonalarterie und damit zu einer Pulmonalinsuffizienz [BA09].

Verwirbelungen im Blutfluss können nicht nur als Symptome einer bestehenden krankhaften Veränderung auftreten, sondern auch selbst weitere Schäden verursachen. So begünstigen die Fluktuationen der Flussgeschwindigkeit innerhalb von Wirbeln die Bildung von Thromben [DKS⁺08], die wiederum andere Blutgefäße verstopfen können. Zudem kann verwirbelter Blutfluss durch seine Scherkräfte hohen Druck auf die Gefäßwand ausüben (*Wall Shear Stress*), der im Verdacht steht, wiederum zur Bildung neuer Aneurysmen beizutragen.

2.1.3 4D PC-MRI

Bei *4D Phase Contrast Magnetic Resonance Imaging* (4D PC-MRI) handelt es sich um eine Bildmodalität, die es erlaubt, den Blutfluss eines Patienten dreidimensional und zeitaufgelöst zu erfassen. Sie basiert darauf, dass sich in Bewegung befindliches Blut innerhalb eines Gradientenfeldes einen Phasenunterschied im Vergleich zu stationärem Gewebe aufbaut, der proportional zur Flussgeschwindigkeit steigt [LMLG02].

Vor der Weiterverarbeitung gemessener Flussdaten sollten durch die MR-Aufnahme entstandene Wirbelstromartefakte wie *Eddy Currents* und *Phase Wraps* entfernt werden. Eddy Currents entstehen durch Änderungen im Magnetfeld des Scanners und äußern sich als langsam ändernde Gradienten, die sich über den Bildinhalt legen. Um diese zu korrigieren, ist es erforderlich, statisches Gewebe in den Aufnahmen zu finden. An solchen Stellen messbarer Fluss ist den Eddy Currents zuzuordnen und wird verwendet, um die Störung zu errechnen [LHM⁺05] und vom Bild zu subtrahieren. Dies ist in [Abbildung 2.2](#) dargestellt.

Phase Wraps treten auf, wenn die gemessene Flussgeschwindigkeit eines Bereichs die bei der Aufnahme am MRT festgelegte Höchstgeschwindigkeit (*velocity encoding sensitivity*, V_{ENC}) überschreitet. Der Wert springt daraufhin auf die andere Seite der messbaren Skala. Bei einem Wertebereich von -1 m/s bis 1 m/s, wobei -1 einen Fluss in eine und 1 einen Fluss in die entgegengesetzte Richtung markiert, würde eine reale Flussgeschwindigkeit von 1.2 m/s in den Daten als -0.8 m/s auftauchen.

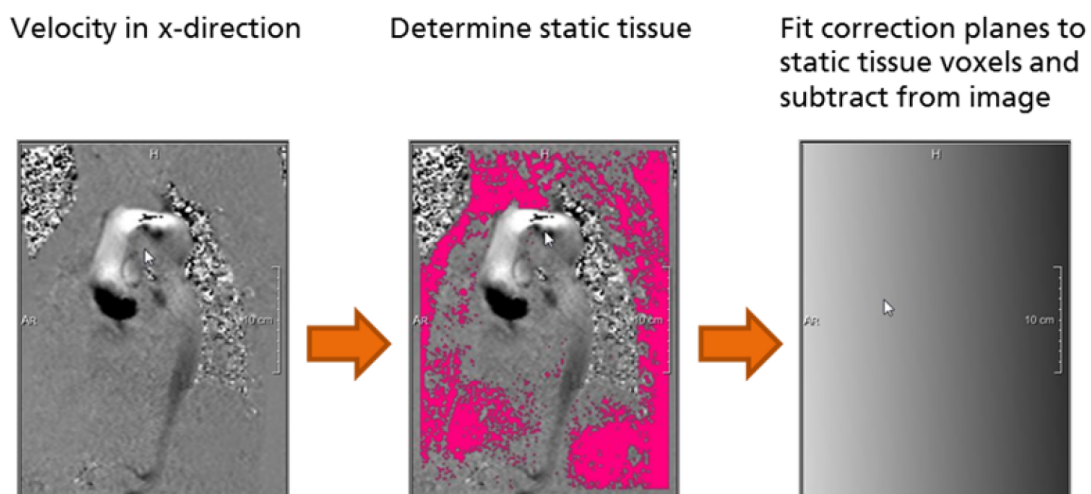


Abbildung 2.2: Ermittlung der Einflüsse der Eddy Currents durch Analyse des statischen Gewebes. [HFS⁺11]

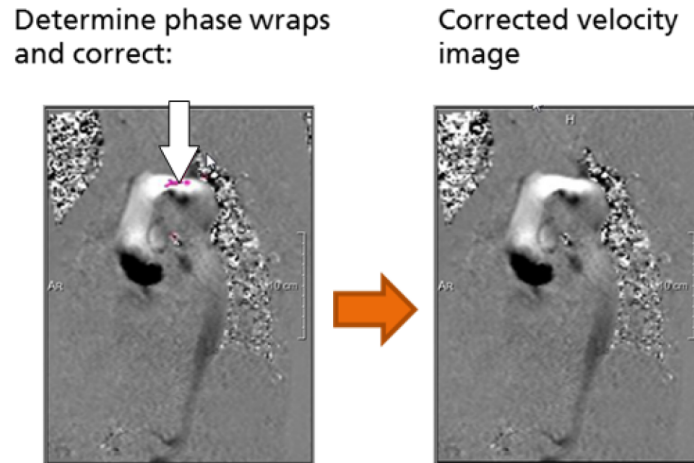


Abbildung 2.3: Detektion und Entfernung von Phase Wraps. [HFS⁺11]

Phase Wraps erscheinen damit als schwarze Flecken in einem ansonsten weißen Bereich oder umgekehrt. Ein Beispiel für deren Detektion und Korrektur ist in [Abbildung 2.3](#) dargestellt.

Als Ergebnis der 4D PC-MRI Untersuchung entstehen vier einzelne, vierdimensionale Bilder. Drei dieser Bilder enthalten die Flusstärke pro Voxel und Zeitschritt jeweils in x -, y - und z -Richtung (*Flow*), das letzte enthält die Flussgeschwindigkeit (*Magnitude*) [HFS⁺11]. Diese Bilder werden zu einem einzigen Vektorfeld zusammengefasst, in dem Flusstärke- und Richtung als dreidimensionale Vektoren gespeichert sind. Zudem wird eine *temporal Maximum Intensity Projection (tMIP)* aus dem Magnitude-Bild errechnet. Hierbei handelt es sich um ein dreidimensionales Bild, das für jede Voxelposition die maximale Flussgeschwindigkeit über alle Zeitschritte enthält. Somit erhalten alle Bereiche des Gefäßes, in denen zu irgend einem Zeitpunkt während der Aufnahme Blut schnell geflossen ist, einen hohen Intensitätswert und grenzen sich damit von der Umgebung des Gefäßes ab.

2.2 Mathematische Grundlagen

2.2.1 Integration in Vektorfeldern

Ein Vektorfeld $\mathbb{V}$ ist eine Funktion, die jedem Element x einer Menge $\Omega \subset \mathbb{R}^n$ einen Vektor $\mathbb{V}(x) \in \mathbb{R}^n$ zuordnet.

2D oder 4D PC-MRI-Daten liegen in Form solcher Vektorfelder vor. Die Werte sind dabei in einem regulären Gitter angeordnet. Um innerhalb eines solchen Gitters Vektoren bestimmen zu können, müssen diese interpoliert werden.

Bei der *Nearest Neighbour* Methode wird zunächst der diesem Punkt am nächsten liegende Gitterpunkt bestimmt und dessen Wert als Interpolationsergebnis verwendet. Diese Methode ist zwar sehr effizient, führt jedoch zu unstetigen Ergebnissen. Gebräuchlicher ist die Anwendung *trilinear Interpolation*, bei der das Ergebnis wie in [Abbildung 2.4](#) durch mehrere lineare Interpolationen aus den 8 angrenzenden Gitterpunkten bestimmt wird.

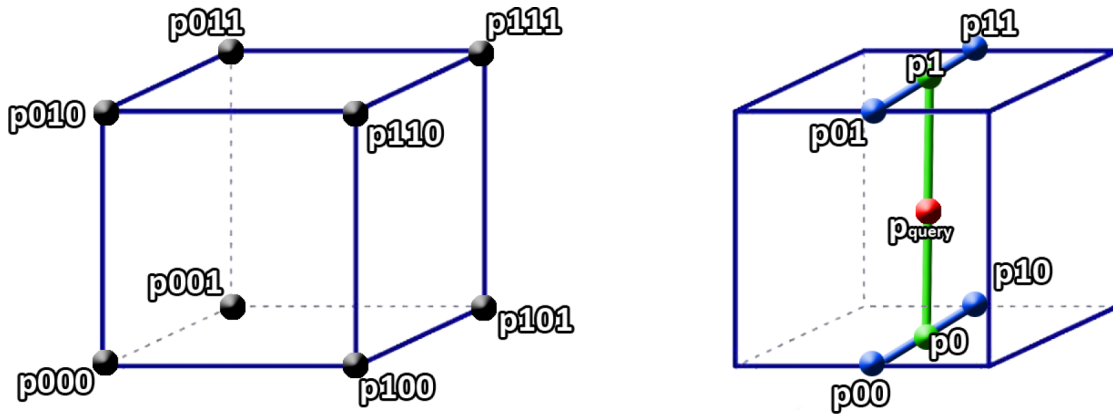


Abbildung 2.4: Trilineare Interpolation. 8 Gitterpunkte (links) werden verwendet, um über mehrere lineare Interpolationsschritte (rechts) den Wert eines beliebigen Punktes p_{query} zu bestimmen. [Köh12]

Die Integration innerhalb eines Vektorfeldes beschreibt das Verfolgen der enthaltenen Vektoren, üblicherweise unter Verwendung einer konstanten Schrittweite s . Das einfachste Verfahren ist die *Euler-Integration*. Hierbei wird der nächste Punkt p_{new} aus der aktuellen Position p_{query} und dem Vektor $\mathbb{V}(p_{query})$ errechnet (siehe [Abbildung 2.5](#)). Dieses Verfahren neigt jedoch dazu, insbesondere bei starken Krümmungen oder hoher Schrittweite, vom tatsächlichen Verlauf abzuweichen. Dieser Fehler akkumuliert sich nach weiteren Integrationsschritten. Zwar kann dies durch eine Verringerung der Schrittweite kompensiert werden, allerdings erhöht dies den Aufwand der Berechnung.

$$p_{new} = p_{query} + s \cdot \mathbb{V}(p_{query}) \quad (2.1)$$



Abbildung 2.5: Euler-Integration: Ein neuer Punkt wird aus dem alten Punkt und dessen Vektor bestimmt. [Köh12]

Eine Alternative stellt die Familie der *Runge-Kutta-Integrationsverfahren* dar. Diese zeichnen sich dadurch aus, dass zusätzlich zu $\mathbb{V}(p_{query})$ noch weitere Vektoren aus der Umgebung von p_{query} für die Integration verwendet werden. Je nach Art des Verfahrens wird dabei eine unterschiedliche Anzahl zusätzlicher Punkte untersucht. Die gebräuchlichsten Varianten RK2 und RK4 verwenden beispielsweise insgesamt zwei bzw. vier Punkte, einschließlich des Startpunkts. Das Ergebnis berechnet sich als gewichtete Interpolation der Samples. Die Funktionsweise beider Verfahren wird schematisch in [Abbildung 2.6](#) dargestellt.

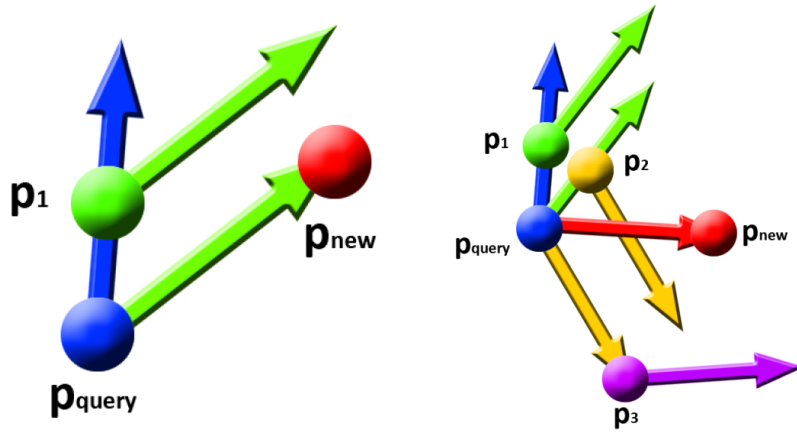


Abbildung 2.6: Visualisierung des Funktionsprinzips von RK2 (links) sowie RK4 (rechts). [Köh12]

Das RK2-Verfahren verwendet dabei folgende Formel, die mit p_{query} und p_1 Vektoren an insgesamt zwei Positionen auswertet.

$$\begin{aligned} p_{new} &= p_{query} + s \cdot \mathbb{V}(p_1) \\ p_1 &= p_{query} + s \cdot \frac{1}{2} \cdot \mathbb{V}(p_{query}) \end{aligned} \quad (2.2)$$

Bei RK4 kommen noch zwei zusätzliche Punkte hinzu. Die Berechnung wird hier folgendermaßen durchgeführt:

$$\begin{aligned} p_{new} &= p_{query} + s \cdot \frac{1}{6} \cdot (\mathbb{V}(p_{query}) + 2 \cdot \mathbb{V}(p_1) + 2 \cdot \mathbb{V}(p_2) + \mathbb{V}(p_3)) \\ p_1 &= p_{query} + s \cdot \frac{1}{2} \cdot \mathbb{V}(p_{query}) \\ p_2 &= p_{query} + s \cdot \frac{1}{2} \cdot \mathbb{V}(p_1) \\ p_3 &= p_{query} + s \cdot \mathbb{V}(p_2) \end{aligned} \quad (2.3)$$

2.2.2 Jacobi-Matrix

Die Jacobi-Matrix J ist eine Matrix bestehend aus allen ersten partiellen Ableitungen einer differenzierbaren Funktion f . Zu ihrer Bildung werden deren Komponentenfunktionen f_1 bis f_m sowie Parameter x_1 bis x_n verwendet:

$$J_f(\vec{x}) = \frac{\partial f(\vec{x})}{\partial \vec{x}} = \begin{pmatrix} \frac{\partial f_1(\vec{x})}{\partial x_1} & \cdots & \frac{\partial f_1(\vec{x})}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m(\vec{x})}{\partial x_1} & \cdots & \frac{\partial f_m(\vec{x})}{\partial x_n} \end{pmatrix} \quad (2.4)$$

Da es sich bei Vektorfeldern ebenfalls um Funktionen handelt, kann auch aus diesen eine Jacobi-Matrix gebildet werden. Die Ableitungen der Komponentenfunktionen werden in diesem Fall mithilfe von Gradienten gebildet:

$$\frac{\partial \mathbb{V}(x_0, \dots, x_i, \dots, x_n)}{\partial x_i} = \frac{\mathbb{V}(x_0, \dots, x_i + s, \dots, x_n) - \mathbb{V}(x_0, \dots, x_i - s, \dots, x_n)}{2s} \quad (2.5)$$

Bei s handelt es sich um die Skalierung, für die üblicherweise die Gitterbreite der Dimension x_i verwendet wird. Dies setzt jedoch voraus, dass das Vektorfeld $\mathbb{V}$ ein regelmäßiges Gitter verwendet. Ist dies nicht der Fall, so bleibt die Ermittlung der Jacobi-Matrix weiterhin möglich, wird jedoch komplexer.

2.3 Technische Grundlagen

2.3.1 Shader-Modell

Shadereinheiten sind auf modernen Grafikchips (*Graphics Processing Unit, GPU*) verbaute Recheneinheiten, welche für die Berechnung von 2D und 3D-Grafiken verwendet werden. Dazu führen sie kleine Programme (sog. *Shader*) aus, welche die gespeicherten Grafikprimitive mit ihren Koordinaten in eine perspektivische, pixelbasierte Darstellung auf dem Bildschirm umwandeln. Diese schrittweise Umwandlung der Grafikprimitive in eine Pixeldarstellung wird als *Shader-Pipeline* bezeichnet und ist in [Abbildung 2.7](#) vereinfacht dargestellt.

Die Recheneinheiten der GPU unterscheiden sich in ihrem Konzept fundamental von der zentralen Recheneinheit (*Central Processing Unit, CPU*) eines Computers. Letztere verfügt über wenige separate Recheneinheiten, die jedoch mit hoher Geschwindigkeit arbeiten und über einen umfangreichen Instruktionssatz verfügen. Die GPU hingegen setzt auf eine hohe Anzahl unabhängiger Recheneinheiten, die jedoch vergleichsweise langsam und mit geringerem Instruktionssatz arbeiten. Dieser Aufbau ist der Funktion der GPU als Grafikverarbeitungseinheit geschuldet, da hier viele einzelne Grafikprimitive unabhängig voneinander verarbeitet werden. Die Stärken der CPU liegen demzufolge bei der sequentiellen Bearbeitung komplexer Programme, während die GPU eher für die parallele Bearbeitung mehrerer kleiner und einfacherer Abläufe geeignet ist.

2.3.1.1 Rendering-Pipeline

Bis zur Einführung von Microsoft DirectX 8.0 [[Zoh11](#)] und OpenGL 2.0 [[Kar09](#)] waren diese Schritte in der Pipeline fest in die Hardware integriert und erlaubten keinerlei Manipulationen. Danach konnten Teile dieser Pipeline durch den Entwickler überschrieben und damit in ihrer Funktionalität erweitert werden. So ist es beispielsweise möglich, anstelle des normalen Vertex-basierten Beleuchtungsmodells (*Gouraud Shading*) das Phong-Beleuchtungsmodell zu implementieren, indem Vertex und Fragment Shader ersetzt werden. Dieses erzeugt eine wesentlich realistischere Beleuchtung von 3D-Modellen, da Beleuchtungseffekte nicht mehr nur für

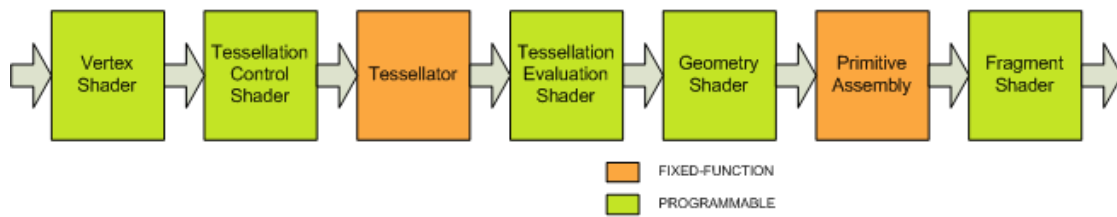


Abbildung 2.7: Shader-Pipeline zur Umwandlung von Grafikprimitiven (links) in Pixel (rechts). [Rák10]

die Eckpunkte der Flächen berechnet und dann interpoliert, sondern für jeden Pixel neu generiert werden. [Abbildung 2.8](#) zeigt den Unterschied zwischen beiden Beleuchtungsverfahren.

Bei der Arbeit mit eigenen Shadern muss beachtet werden, dass diese die vorgegebenen Varianten aus der Pipeline vollständig ersetzen. Es ist demzufolge erforderlich, nicht nur die gewünschte erweiterte Funktionalität eines Shaders zu implementieren, sondern auch dessen Grundfunktionalität.

Die Aufgabe des Vertex Shaders ist es, die Position eines Eckpunkts unter Betrachtung von Kameraposition, Blickrichtung und perspektivischer Verzerrung vom Welt- in das Kamerakoordinatensystem zu überführen. Der Entwickler hat durch diesen Shader die Möglichkeit, Positionen von Eckpunkten zu manipulieren und somit die Objektform zu ändern. Auch kann er zusätzliche Informationen an jeden Punkt anhängen, die von späteren Shadern ausgelesen werden können.

Die nachfolgenden Tessellation- sowie der Geometry-Shader führen von sich aus keine Operationen durch. Der Geometry Shader ermöglicht es, nicht nur Eckpunkte anders zu platzieren (wie der Vertex Shader), sondern auch neue Eckpunkte zu erzeugen oder bestehende Punkte zu entfernen. Dies kann u.a. dazu verwendet werden, die Komplexität von Objekten in Abhängigkeit von der Entfernung zur Kamera automatisch zu reduzieren, um die nachfolgenden Shader zu entlasten. Der Tessellation-Shader hat eine ähnliche Funktion, dient aber primär dazu, den Detailgrad von Objekten nachträglich zu erhöhen. Im Gegensatz zum Geometry Shader ist hier die Anzahl neuer Eckpunkte, die erzeugt werden können, praktisch unbegrenzt.

Die Farbe eines Pixels auf dem Bildschirm wird vom Fragment Shader bestimmt. Dabei werden u.a. Farb- und Texturwerte der Eckpunkte, die vom Vertex oder Geometry Shader geliefert werden, für jeden Punkt von Flächen oder Linien interpoliert.

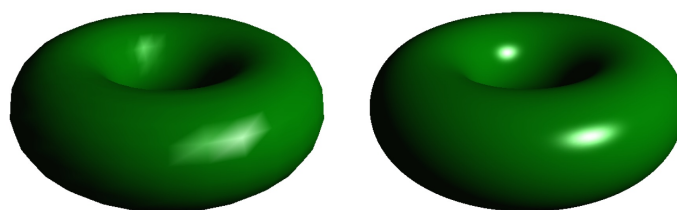


Abbildung 2.8: Vergleich der Beleuchtungseffekte zwischen *Gouraud Shading* (links) und *Phong Shading* (rechts). [Sch10]

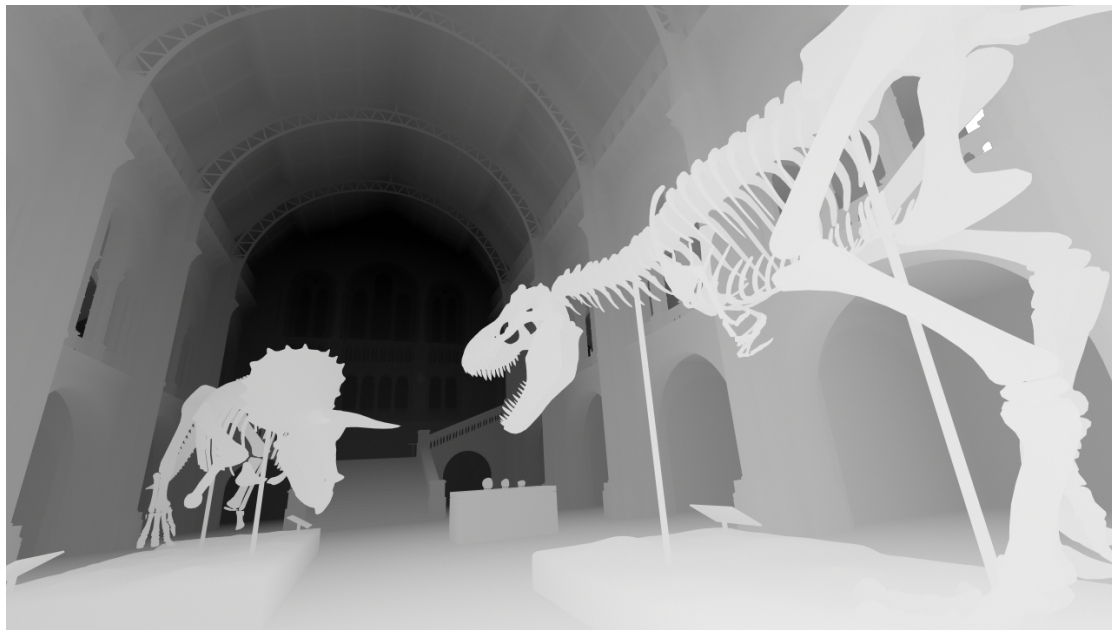


Abbildung 2.9: Tiefenwerte einer 3D-Szene im Z-Buffer. Nahe Objekte erscheinen hell, weiter entfernte dunkel. Quelle: Hongyuan Jia³

Zudem findet hier auch der Tiefentest statt, der dafür sorgt, dass verdeckte Objekte nicht auf dem Bildschirm angezeigt werden. Als ein Nebenprodukt des Tiefentests entsteht der sog. Z-Buffer - eine Textur, die für jeden Bildpixel die Entfernung des darauf abgebildeten 3D-Objekts zur Kamera speichert. Ein Beispiel für das Aussehen eines Z-Buffers findet sich in [Abbildung 2.9](#).

2.3.1.2 Multipass-Rendering

Für viele grafische Effekte ist es erforderlich, das Rendering mehrfach mit unterschiedlichen Parametern, Kameraeinstellungen sowie Shadern durchzuführen und die Ergebnisse in einem einzelnen Bild zu kombinieren (sog. *Multipass-Rendering*). Dafür kann der Fragment Shader so modifiziert werden, dass er sein Ergebnis nicht auf den Bildschirm, sondern in ein *Shader Storage Buffer Object (SSBO)* oder über ein *Frame Buffer Object (FBO)* in eine Textur schreibt. Auf diese kann nun in folgenden Rendering-Vorgängen zugegriffen werden.

Um die Ergebnisse mehrerer Rendering-Vorgänge zusammenzuführen oder Bildeffekte wie Unschärfe auf den gesamten Bildinhalt anzuwenden, wird ein sog. *Screenfilling Quad* gerendert, wobei es sich um ein einfaches, den Bildschirm komplett füllendes Grafikprimitiv handelt. Der Fragment Shader wird nun für jeden einzelnen Pixel des Bildes aufgerufen und kann Werte aus einer oder mehreren Texturen beziehungsweise SSBOs kombinieren.

2.3.1.3 General Purpose GPU

Obwohl die Grafikkarte primär für die Berechnung von 2D und 3D-Grafiken gedacht ist, ermöglichen Shader auch allgemeine Berechnungen (*General Purpose*) unabhängig von der Rendering-Pipeline. Dafür gibt es unterschiedliche Ansätze. So ist es

³<http://people.cs.clemson.edu/~hongyuj/809/hw4/zBuffer.jpg>

beispielsweise möglich, die in der Rendering Pipeline verwendeten Shader so zu programmieren, dass sie statt einer Grafikausgabe eine andere Berechnung durchführen.

Der Grafikkarten-Hersteller *nVidia* bietet mit seinem System CUDA eine Schnittstelle an, durch die Teile eines Programms ohne direkte Programmierung von Shadern auf die GPU ausgelagert werden können. Da CUDA populäre Programmiersprachen wie *Java*, *C++*, *C#* und *Python* unterstützt, muss keine Shader-Sprache wie *GLSL* erlernt werden [NVI14]. Eine freie Alternative zu nVidias proprietärer CUDA-Plattform ist die *Open Computing Language (OpenCL)*, die auch mit Grafikkarten anderer Hersteller eingesetzt werden kann.

Seit Version 4.3 existieren in OpenGL sog. *Compute Shader*, die für die Durchführung von GPGPU-Berechnungen vorgesehen sind [Khr15]. Eine beliebige Anzahl dieser Compute Shader kann in mehreren Gruppen innerhalb eines mehrdimensionalen Gitters instantiiert werden. Jede Instanz erhält als Eingabewerte Informationen über ihre Position im Gitter, aus der eine globale, einzigartige ID errechnet werden kann. Mit diesen Informationen kann die Instanz beispielsweise feststellen, in welchem Speicherbereich sie arbeiten soll.

2.3.2 Indirect Volume Rendering

Aus 4D PC-MRI-Daten lassen sich skalare Volumendaten berechnen, die durch eine Reihe von Verfahren visualisiert werden können. Häufige Anwendung findet dabei das *indirekte Volumenrendering*. Dieses basiert darauf, Oberflächen mit gleicher Intensität (*Iso-Oberflächen*) innerhalb der Datensätze zu finden und in ein Dreiecksnetz umzuwandeln, welches dann wiederum gerendert werden kann.

Das Finden der Oberfläche sowie Erstellen des Dreiecksnetzes kann auf verschiedene Weisen erfolgen. Eine Möglichkeit ist die Anwendung eines Iso-Wertes. Ist die Trennung zwischen Objekt und Hintergrund durch einen einzelnen Iso-Wert nicht möglich, so kann der Datensatz segmentiert werden. Auch hierfür gibt es verschiedene Möglichkeiten, beispielsweise Wasserscheidentransformation, Region Growing oder das Einzeichnen von Begrenzungslinien in einzelnen Schichtbildern. Letztendlich entsteht jedoch immer eine Segmentierung, die jeden Voxel des Datensatzes entweder Vorder- oder dem Hintergrund zuordnet. Mithilfe der Segmentierung oder eines Iso-Wertes kann die Oberfläche des Objekts in ein Dreiecksnetz konvertiert werden, beispielsweise durch Anwendung des *Marching-Cubes*-Verfahrens.

Andere Methoden zum Erzeugen des Dreiecksnetzes erfordern keine Segmentierung, sind jedoch weiterhin von Nutzereingaben abhängig. Ein Beispiel hierfür sind *aktive Konturen*, die vom Anwender eingezeichnet werden und daraufhin mithilfe einer Optimierungsfunktion (beispielsweise auf Basis von Gradienten) selbstständig zu einer Oberfläche konvergieren. Unabhängig vom gewählten Verfahren entsteht am Ende ein Dreiecksnetz der Oberfläche.

Ein Vorteil dieses Vorgehens ist, dass das erzeugte Dreiecksnetz direkt von der GPU visualisiert werden kann. Auch lassen sich viele Effekte wie Beleuchtung und realistischer Schattenwurf sehr einfach umsetzen. Zudem ist es möglich, einzelne Objekte gezielt ein- oder auszublenden. Auch quantitative Untersuchungen wie die Messung von Entfernungen oder Krümmungen lassen sich mithilfe der Dreiecksnetze gut

durchführen. Eine Segmentierung vereinfacht zudem Interaktionen wie beispielsweise das Auswählen einzelner Objekte.

Ein Nachteil ist, dass bei einer Anpassung der Segmentierung oder des Iso-Wertes der *Marching Cubes*-Algorithmus erneut durchgeführt werden muss. Dies stellt insbesondere bei der Exploration von Datensätzen ein Problem dar, da kein sofortiges Feedback auf eine Änderung der Segmentierung oder des Iso-Wertes erfolgen kann. Auch lassen sich nur solche Objekte durch Dreiecksnetze beschreiben, die eine eindeutige Oberfläche besitzen. Bei amorphen Strukturen wie fließenden Gewebeübergängen oder Gasen lässt sich eine Oberfläche jedoch nicht eindeutig festlegen.

2.3.3 Direct Volume Rendering

Ein grundsätzlich anderes Verfahren zur Visualisierung skalarer Volumendaten ist das *Direct Volume Rendering* (DVR). Im Gegensatz zum *Indirect Volume Rendering* ist bei DVR die Bildung einer Iso-Oberfläche und eines damit verbundenen Dreiecksnetzes nicht erforderlich. Stattdessen werden die Bildintensitäten über eine Transferfunktion in Farbinformationen umgewandelt und direkt auf den Bildschirm projiziert [DCH88].

Verfahren, um DVR durchzuführen, lassen sich grundlegend in die drei Kategorien einteilen:

- *Bildbasierte Verfahren* iterieren über alle Pixel des Ausgabebildes und berechnen, welche Voxel des Datensatzes auf diesen abgebildet werden.
- *Objektbasierte Verfahren* hingegen iterieren über die Voxel des Datensatzes und ermitteln, auf welche Pixel des Ausgabebildes diese abgebildet werden.
- *Texturbasierte Verfahren* basieren auf dem Ansatz, durch das Rendern einzelner Schichtbilder in Form von Texturen einen dreidimensionalen Effekt zu erzeugen, über den der komplette Datensatz visualisiert wird. [Abbildung 2.10](#) zeigt das grundlegende Prinzip hinter diesem Verfahren.

In dieser Arbeit wird *GPU-Raysampling* aus der Gruppe der bildbasierten Verfahren eingesetzt. Hierbei können die Pixel des Ausgabebildes voneinander unabhängig berechnet werden, wodurch das Verfahren hochgradig parallelisierbar und damit perfekt für die Berechnung auf der Grafikkarte geeignet ist. Durch die rasante Entwicklung von Grafikprozessoren in den letzten Jahren ist GPU-Raysampling heute das Standardverfahren für DVR.

Die Grundidee dabei ist es, Strahlen vom Kamerapunkt durch die Bildebene zu senden. Auf Bereichen der Strahlen, die das Volumen schneiden, werden in regel- oder unregelmäßigen Abständen Sample-Punkte definiert, an denen die entsprechenden skalaren Datenwerte des Volumens interpoliert werden. Die aufeinander folgenden Datenwerte werden mithilfe der Transferfunktion in Farb- sowie Transparenzwerte umgerechnet und akkumuliert. Die resultierende Farbe wird in das berechnende Pixel eingetragen.

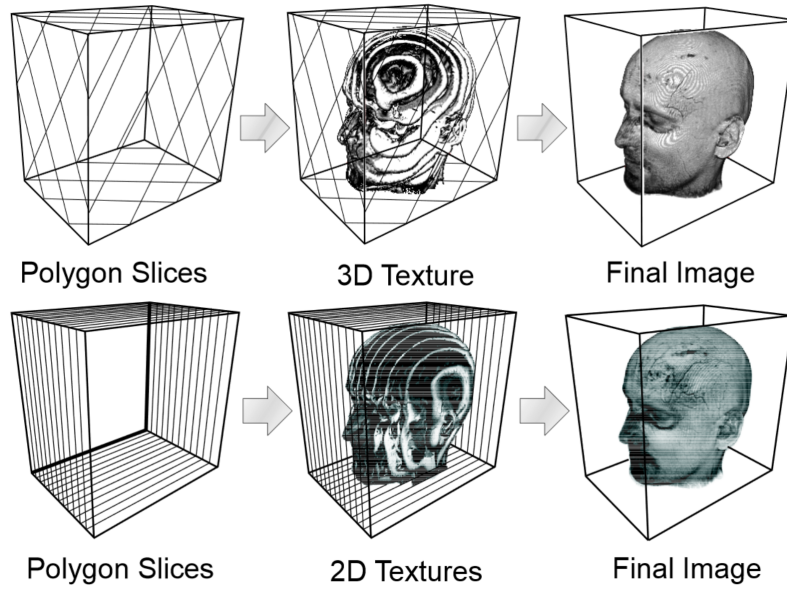
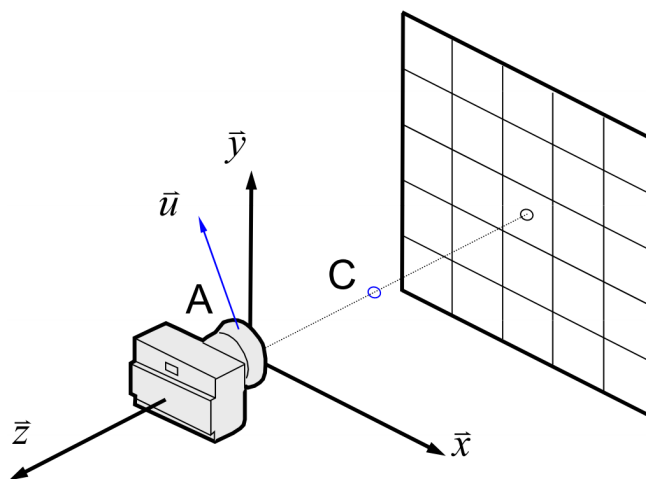


Abbildung 2.10: Texturbasiertes DVR; Die Visualisierung wird durch das Überlagern einzelner Texturschichten erzeugt, die entlang der Blickrichtung (*view-aligned*, oben) oder der Koordinatenachsen (*axis-aligned*, unten) ausgerichtet sind [EE02].

Ein Kernproblem bildbasierter Verfahren ist, den Ein- sowie Austrittspunkt jedes Strahls in und aus dem 3D-Volumen basierend auf der Kameraposition und Blickrichtung zu ermitteln. Dazu ist es zunächst nötig, die Richtung der Strahlen durch die Bildebene zu bestimmen, um damit die Schnittpunkte mit dem Datenvolumen zu ermitteln. Eine klassische Methode hierfür ist die direkte Berechnung unter Verwendung der Kameraposition A , eines Blickpunktes C sowie des *Up-Vektors*, der die Rotation der Kamera bestimmt. Mithilfe dieser Werte können zunächst die Koordinatenachsen $\vec{x}$, $\vec{y}$ und $\vec{z}$ berechnet werden [Gro10]:



$$\begin{aligned}
 \vec{z} &= \frac{A - C}{|A - C|} \\
 \vec{x} &= \frac{\vec{u} \times \vec{z}}{|\vec{u} \times \vec{z}|} \\
 \vec{y} &= \vec{z} \times \vec{x}
 \end{aligned} \tag{2.6}$$

Abbildung 2.11: Darstellung der Bestimmung der Koordinatenachsen für die Berechnung des Strahls [Gro10].

Wie in [Abbildung 2.11](#) dargestellt, verläuft die Bildebene (*near-Plane*) parallel zu der von $\vec{x}\vec{y}$ aufgespannten Ebene. Unter Verwendung von Höhe h und Breite b sowie des Öffnungswinkels der Kamera θ können nun der Ursprungspunkt der Bildebene O sowie die Vektoren $\vec{x}'$ und $\vec{y}'$ berechnet werden, mit denen über die Pixel der Bildebene iteriert werden kann [[Gro10](#)].

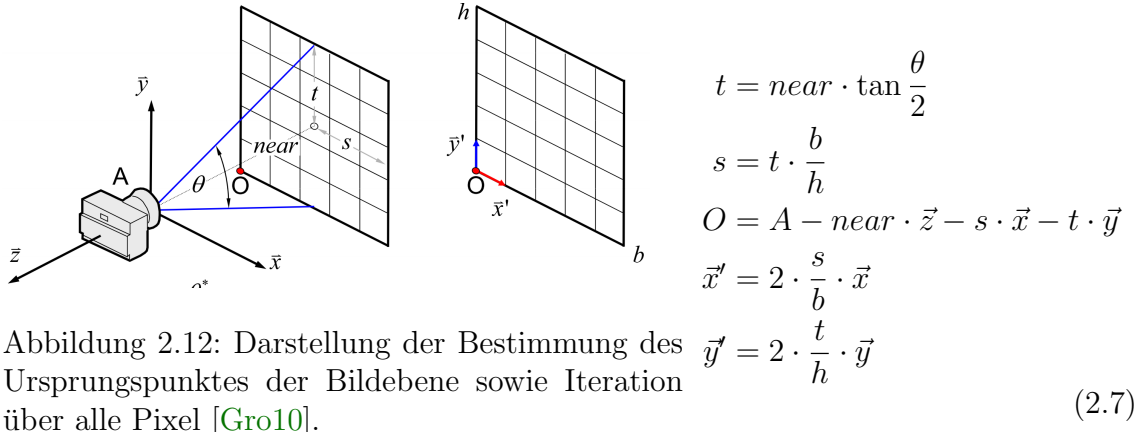


Abbildung 2.12: Darstellung der Bestimmung des Ursprungspunktes der Bildebene sowie Iteration über alle Pixel [[Gro10](#)].

Da der Ursprung der Bildebene O nun bekannt ist und die Längen der Vektoren $\vec{x}'$ und $\vec{y}'$ der Breite bzw. Höhe eines Pixels entsprechen, können nun Strahlen für beliebige Pixel berechnet werden.

Die manuelle Berechnung der Strahlen sowie Ermittlung der Schnittpunkte mit dem Datenvolumen kann man jedoch unter Verwendung von Funktionen der Grafikhardware umgehen, indem man einen *Farbwürfel* verwendet [[KW03](#)], dessen Seitenverhältnisse den Dimensionen des zu visualisierenden Datensatzes entspricht. Jeder Ecke dieses Würfels wird eine Farbe entsprechend ihrer Position zugewiesen. So wird die Ecke des Würfels im Ursprung des Koordinatensystems schwarz gefärbt, da sie die Position $(0, 0, 0)$ einnimmt. Die Ecke mit der Position $(1, 0, 0)$ wird rot gefärbt, diejenige mit der Position $(1, 1, 1)$ weiß etc. Die Farbe jedes Punktes auf den Flächen des Würfels wird aus den anliegenden 4 Eckpunkten interpoliert.

Das so entstandene Bild wird zweimal gerendert, das Ergebnis wird jeweils in einer Textur oder einem Buffer zwischengespeichert. Beim ersten Durchgang werden nur die der Kamera zugewandten Seiten des Würfels gezeichnet (*Backface-Culling*), beim zweiten die jeweils anderen Seiten (*Frontface-Culling*). Das Ergebnis dieses Renderrings ist in [Abbildung 2.13](#) dargestellt. Pro Pixel entstehen somit zwei Farbwerte - je einer von der Vorder- und Hinterseite des Würfels. Diese können als Eingangs- und Ausgangspunkt des Strahls aus dem 3D-Volumen interpretiert werden, der sich wiederum durch die Differenz beider Punkte ergibt.

DVR kann in Echtzeit berechnet und über eine Manipulation der Transferfunktion auch angepasst werden. Eine tiefergehende Betrachtung der Transferfunktionen und ihrer Erzeugung findet sich in [Abschnitt 2.3.4](#).

Nach Anwenden der Transferfunktion verfügt jeder Punkt auf dem Strahl über einen Farb- und einen Opazitätswert. In dieser Arbeit werden die Bildintensitätswerte jedoch nur auf Grauwerte abgebildet. Diese werden zusammen mit den Opazitätswert unter Erhaltung ihrer Tiefeninformationen akkumuliert. Dies kann bereits während

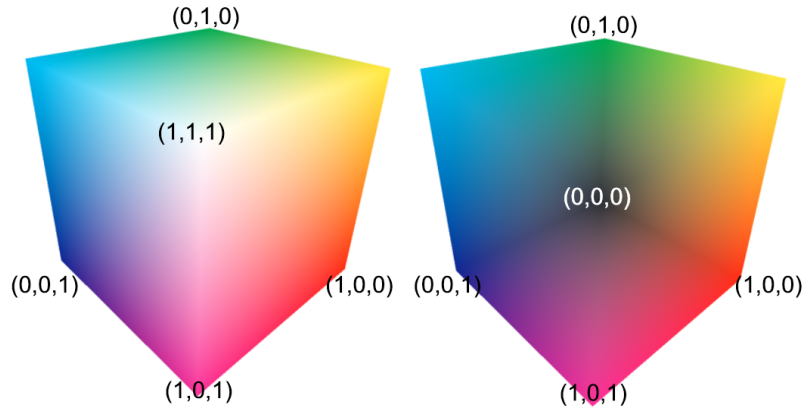


Abbildung 2.13: Rendering des Farbwürfels: Backface-Culling (links) und Frontface-Culling (rechts) [KW03]

der Iteration über den Strahl geschehen. Somit müssen nur die akkumulierten Grau- und Transparenzwerte anstelle aller ausgelesenen Intensitätswerte gespeichert werden.

Die an jedem Sample-Punkt s_p erzeugten Opazitätswerte o_p sowie Grauwerte g_p werden mit den bereits akkumulierten Werten für Opazität O_{p-1} und Grauwert G_{p-1} verrechnet. Hierbei werden die Startwerte $O_0 = 1$ und $G_0 = 0$ verwendet. Die Akkumulation der Opazität dient lediglich der korrekten Berechnung des Grauwerts; der entstehende Pixel hat grundsätzlich einen Opazitätswert von 1.

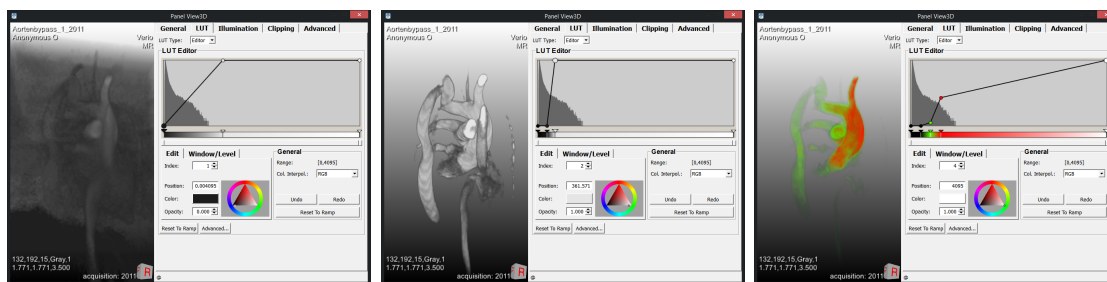
$$O_p = O_{p-1} \cdot (1 - o_p) \quad G_p = G_{p-1} + O_p \cdot g_p \quad (2.8)$$

Raysampling kann beschleunigt werden, indem die Berechnung abgebrochen wird, wenn die akkumulierten Opazitätswerte einen festgelegten Wert überschreiten. Üblicherweise wird hier ein Wert nahe 1 gewählt, da nachfolgende Punkte auf dem Strahl nur noch einen minimalen Einfluss auf das Endergebnis haben. Dies wird als *Early Ray Termination* bezeichnet.

Eine Möglichkeit, DVR ohne eine solche Akkumulation einzusetzen, ist mittels *Maximum Intensity Projection (MIP)*. Hier wird lediglich der höchste Datenwert pro Strahl als Grauwert dargestellt. Bei diesem Vorgang gehen sämtliche Tiefeninformationen verloren, was jedoch durch die Möglichkeit, die Szene zu rotieren, ausgeglichen werden kann.

Ein Nachteil des DVR ist, dass keine Beleuchtung zur Verfügung steht, da hier keine Oberflächennormalen vorhanden sind. Als Ersatz können Gradienten verwendet werden, da diese wie Oberflächennormalen senkrecht auf starken Grauwert-Übergängen stehen, die in der Visualisierung als Oberflächen wahrgenommen werden.

Essentiell für die Qualität der Visualisierung ist die Abtastrate des Strahls. Je öfter der Strahl abgetastet wird, desto mehr Informationen liegen für die Visualisierung vor. Allerdings erfordert eine höhere Abtastrate auch mehr Leistung. Nach dem *Nyquist-Shannon-Abtasttheorem* sollte der Abstand der Sample-Punkte nicht größer sein als die Hälfte der Voxelausdehnung, um die Artefaktbildung zu vermeiden [Nyq28].



(a) Schlecht eingestellte Transferfunktion

(b) Transferfunktion mit Abbildung auf Transparenz und Hel-
ligkeit(c) Transferfunktion mit zu-
sätzlicher Abbildung auf Farbe

Abbildung 2.14: Grafischer Transferfunktionen-Editor aus MeVisLab mit verschiedenen eingestellten Transferfunktionen. Die Visualisierung zeigt Aorta und Pulmonalarterie.

2.3.4 Transferfunktionen

Die Auswahl der richtigen Transferfunktion ist essentiell beim DVR. Bereiche im Datensatz, die nicht Teil der zu untersuchenden Objekte sind, sollten dabei möglichst transparent dargestellt werden. Der linke Teil von [Abbildung 2.14](#) zeigt ein Beispiel einer schlecht gewählten Transferfunktion. Das umgebende Gewebe ist deutlich zu sehen und schwer vom eigentlichen Fokusobjekt zu unterscheiden.

Die Transferfunktion sollte so angepasst werden, dass die Anatomie gut sichtbar und vom umliegenden Gewebe abgegrenzt ist. Bei CT-Daten ist es möglich, Transferfunktionen vollautomatisch basierend auf Hounsfield Units zu generieren. Diese beschreiben die Strahlenabsorption von Stoffen sowie Gewebetypen und sind konstant zwischen verschiedenen Datensätzen.

Dies ist bei MRT-Daten nicht möglich, da hier bestimmten Intensitätswertebereichen nicht eindeutig und geräteübergreifend Materialien zugeordnet werden können. Demzufolge ist eine Anpassung der Transferfunktion durch den Anwender erforderlich. Diese kann vollständig manuell oder semi-automatisch erfolgen. Bei einer manuellen Anpassung ist es üblich, den Funktionsgraphen der Transferfunktion wie in [Abbildung 2.14](#) über die Benutzeroberfläche direkt zu manipulieren. Die Einblendung eines Histogramms im Hintergrund ermöglicht dabei eine bessere Orientierung. In der semi-automatischen Variante werden dem Nutzer automatisch generierte Transferfunktionen mit einem Screenshot der daraus entstehenden Visualisierung angeboten, von denen er sich diejenige aussuchen kann, die seinen Anforderungen am besten entspricht. Um die gewählte Funktion weiter anzupassen, kann dieser Schritt mehrmals wiederholt werden. [Abbildung 2.15](#) zeigt eine Benutzeroberfläche, die der Auswahl einer von mehreren automatischen Transferfunktionen dient.

Unter Hinzunahme weiterer Informationen, wie beispielsweise Gradienten, lassen sich mehrdimensionale Transferfunktion anwenden. Zweidimensionale Transferfunktionen werden als Flächen in einem ebenfalls zweidimensionalen Histogramm dargestellt. Ein Beispiel dafür ist in [Abbildung 2.16](#) zu sehen. Bei der Verwendung von Gradienten zeigen sich Objektgrenzen im Histogramm durch charakteristische Bogenstrukturen, die auch in der Abbildung erkennbar sind. Unter guten Bedingun-

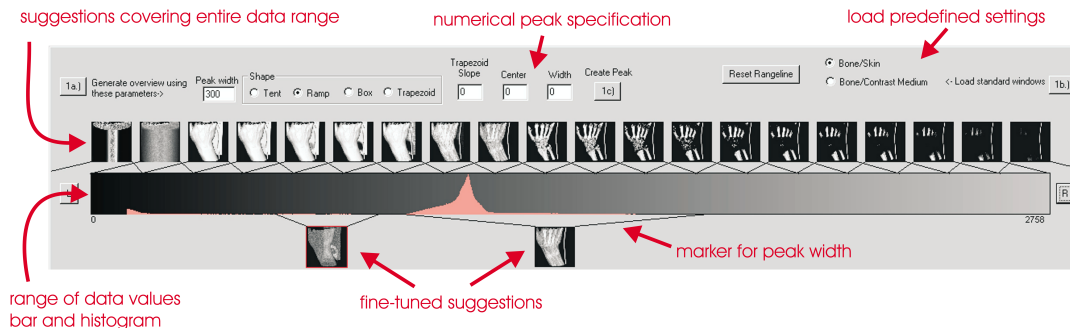


Abbildung 2.15: Bestimmung einer Transferfunktion durch automatische Generierung verschiedener Funktionen und Auswahl der besten Funktion durch den Nutzer. [KG01]

gen lassen sich diese als Ansatzpunkte für die Positionierung der Transferfunktion verwenden [KD98], allerdings ist ihre Sichtbarkeit stark von der Qualität der Datensätze abhängig. So führt beispielsweise schon geringes Rauschen zu einem starken Verschwimmen der Bögen. Über dem Bild liegende Gradienten, wie sie beispielsweise von Eddy Currents hervorgerufen werden, können die Bogenstruktur sogar komplett verschwinden lassen. Abhängig von den Intensitätswerten der entsprechenden Objekte ist auch eine Überlappung der Bögen möglich, was eine exakte Selection erschwert [SVSG06]. Abbildung 2.17 zeigt den Einfluss verschiedener Bildstörungen auf das Histogramm.

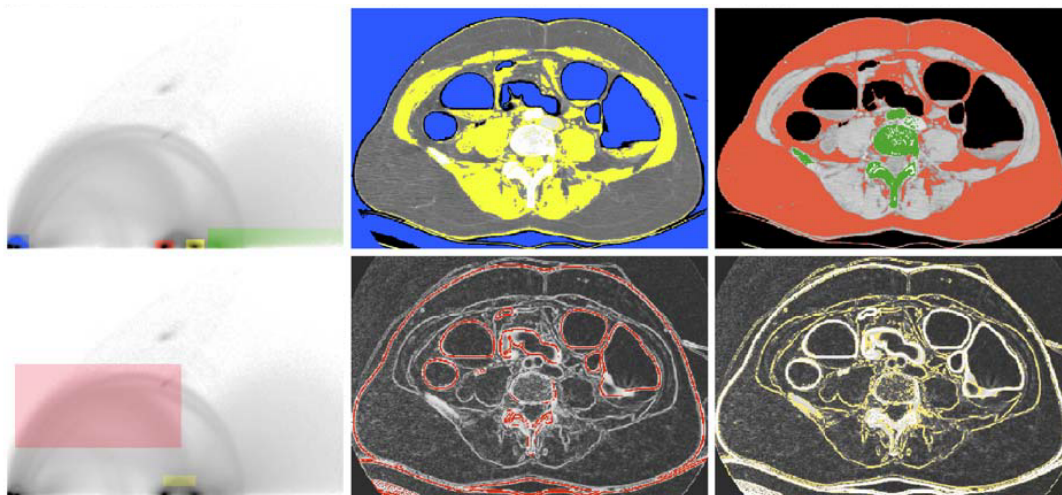


Abbildung 2.16: Zweidimensionale Transferfunktionen auf einem ebenfalls zweidimensionalen Histogramm sowie deren Auswirkungen auf die Visualisierung [SKGM03].

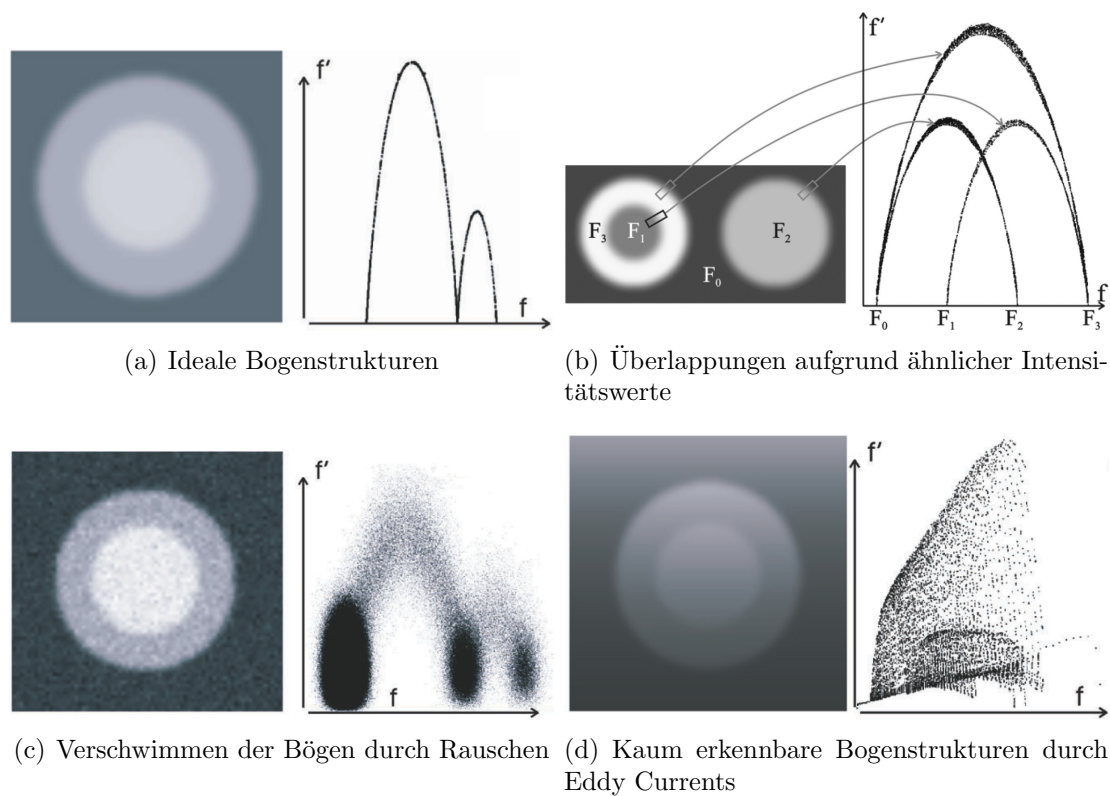


Abbildung 2.17: Verschiedene Störeinflüsse auf zweidimensionalen Histogrammen, die Intensität und Gradienten darstellen [SVSG06].

3. Verwandte Arbeiten

Dieses Kapitel enthält einen Überblick über medizinische Arbeiten, die sich mit der Korrelation zwischen verschiedenen Pathologien des kardiovaskulären Systems spezifischen Blutflussmustern sowie dem Einsatz von 4D PC-MRI zur Untersuchung des Blutflusses beschäftigen. Auch werden hier Arbeiten aufgelistet, welche die Flussvisualisierung sowie die Erkennung von Verwirbelungen behandeln. Dabei werden sowohl die in dieser Arbeit verwendeten Verfahren vorgestellt, als auch alternative Methoden mit ihren Vor- und Nachteilen beschrieben.

3.1 Klinische Studien mittels 4D PC-MRI

Allen et al. prüften die klinische Einsetzbarkeit von 4D PC-MRI Untersuchungen sowie den dadurch gewonnenen Mehrwert im Vergleich zur klassischen 2D PC-MRI Untersuchung [ABP⁺13]. An 16 Patienten mit verschiedenen kardiovaskulären Erkrankungen wurden sowohl 2D als auch 4D PC-MRI-Untersuchungen durchgeführt, anhand derer quantitative Werte wie Flussgeschwindigkeiten oder Rückflussmengen ermittelt wurden. Hier zeigte sich eine durchschnittliche Abweichung von 35% zwischen beiden Verfahren. Auf einer Relevanz-Skala von 1 (niedrig) bis 4 (hoch) wurde die 4D-Untersuchung sowie eine Kombination aus 4D und 2D mit 2.93 bzw. 2.8 bewertet. Allen et al. schlussfolgern daraus, dass 4D PC-MRI-Untersuchungen das Potential haben, akkurate quantitative Analysen zu ermöglichen und 2D-Untersuchungen im klinischen Alltag zu ersetzen.

Viele Studien beschäftigen sich mit Korrelationen zwischen Pathologien und bestimmten Blutflussmustern. Durch das Vorhandensein von Wirbeln lässt sich jedoch nicht zweifelsfrei auf eine Erkrankung schließen. Auch bei gesunden Probanden treten an einigen Stellen Verwirbelungen im Blutfluss auf. In einer Studie von Fredriksson et al. zeigen auch gesunde Probanden beispielsweise Verwirbelungen in der rechten Herzkammer zu Beginn und am Ende der Diastole [FZE⁺11]. Frydrychowicz et al. zeigten, dass Zahl und Stärke natürlicher Verwirbelungen mit dem Alter der Probanden und dem Durchmesser der absteigenden Aorta positiv korreliert [FBM⁺12]. Des Weiteren wird die Frage aufgeworfen, ob die Vergrößerung des Durchmessers

durch die steigende Zahl an Wirbeln bedingt wird oder umgekehrt. Das Vorhandensein natürlicher Wirbel in der Pulmonalarterie wurde von Bächler et al. dokumentiert [BPS⁺13]. Dazu wurden 18 gesunde Probanden mittels 4D PC-MRI untersucht, von denen 15 rechtsdrehenden helikalen Fluss in der rechten Pulmonalarterie aufwiesen.

Dennoch werden verschiedene kardiovaskuläre Erkrankungen mit Veränderungen des Blutflussmusters, insbesondere dem Auftreten von Wirbeln, in Verbindung gebracht. So zeigten François et al., dass eine Fallot-Tetralogie deutliche Auswirkungen auf den Blutfluss in verschiedenen Bereichen des kardiovaskulären Systems hat [FSS⁺12]. Hier wurden Blutflussmuster von 11 Fallot-Patienten mit denen von 10 gesunden Probanden verglichen. In der Mehrzahl der untersuchten Patienten traten demnach verstärkt Verwirbelungen in der rechten Herzkammer sowie der Pulmonalarterie auf.

Ein ähnliches Bild zeigte sich auch bei von Frydrychowicz et al. durchgeführten Untersuchungen [FMH⁺11], bei denen der Blutfluss in der Aorta von 28 ISTA-Patienten mit dem gesunder Probanden verglichen wurde. Eine verstärkte Wirbelbildung in der Aorta wurde bei 25 der Patienten, aber nur fünf der Probanden festgestellt. Zudem wiesen 18 der Patienten Verwirbelungen in den Abzweigungen des Aortenbogens auf, während diese nur bei zwei Probanden auftraten. Bei 19 der Patienten wurden zudem in Bereichen hinter der Stenose Dilatationen des Gefäßes beobachtet. Es wird der Verdacht geäußert, dass deren Entstehung in Zusammenhang mit durch die Flussveränderung entstehendem Wall Shear Stress stehen könnte.

Einen Zusammenhang zwischen dem Aortendurchmesser und dem Auftreten helikaler Verwirbelungen beobachteten Markl et al. in einer Untersuchung mit Marfan-Patienten [MGH⁺11]. Hier wurden drei Personengruppen untersucht, wobei die erste aus 26 Marfan-Patienten, die zweite aus 7 Patienten mit Verdacht auf das Marfan-Syndrom und die dritte aus 10 gesunden Probanden bestand. Bei den Marfan-Patienten lag der durchschnittliche Durchmesser der Aorta um 3 bis 25% über dem der Kontrollgruppe. In der absteigenden Aorta zeigten sich bei diesen Patienten verstärkter helikaler Fluss, bei einigen kam es auch in den Abzweigungen des Aortenbogens zu Verwirbelungen.

Hope et al. verglichen die Flussmuster von Patienten mit einer bikuspiden Aortenklappe mit denen gesunder Probanden [HHM⁺10]. Bei 15 der 20 Patienten wurden helikale Verwirbelungen in der absteigenden Aorta während der Systole entdeckt, die bei den Probanden nicht auftraten. Die Drehrichtung der Verwirbelungen war dabei abhängig davon, welche der Lappen zusammengewachsen waren. Rechtsdrehender helikaler Fluss trat verstärkt bei Patienten mit verbundenen rechten und linken Lappen auf, während linksdrehender Fluss mit einer Fusion von rechtem und zur Herzkammerscheidewand gehörendem Lappen in Verbindung gebracht wurde.

Zu einem ähnlichen Ergebnis kamen auch Meierhofer et al. [MSL⁺13]. Auch hier wurde der Blutfluss von Patienten mit bikuspider Aortenklappe mit dem gesunder Probanden verglichen. Zusätzlich zum Blutfluss wurde auch der Wall Shear Stress untersucht. Im Vergleich mit den Probanden zeigten sich bei Patienten mit bikuspidaler Klappe neben den veränderten Flussmustern mit verstärkter Verwirbelung auch Unterschiede beim Druck auf die Gefäßwand. Axialer Druck in der Aorta war bei den Patienten vermindert, zirkumferenzieller Druck stark erhöht. Der gesamte Druck auf die Gefäßwand war ebenfalls höher als bei gesunden Probanden. Meierhofer et al.

stellten zudem fest, dass Bereiche mit besonders hohem Druck sich im Wesentlichen in der aufsteigenden Aorta befinden, wo bei BAV-Patienten häufig Dilatationen zu beobachten sind.

3.2 Visualisierung von Flussdaten

Eine Vielzahl von Arbeiten beschäftigen sich mit der Visualisierung von Strömungen und Flussbildern in Form von Vektorfeldern. Ansätze zur Visualisierung von Flussdaten lassen sich nach Post et al. grundlegend in direkte oder globale, texturbasierte, geometriebasierte und feature-basierte Techniken einteilen [PVH⁺03].

Ein Beispiel für globale Verfahren ist das von Van Pelt et al. angewandte Raytracing [vOB⁺10a]. Hier wird der Fluss direkt mithilfe von DVR visualisiert, Features wie Flussstärke oder Richtung werden auf Farben abgebildet. Allerdings sind innenliegende Flussstrukturen bei dieser Art der Flussvisualisierung häufig schlecht sichtbar.

Texturbasierte Verfahren bilden Flussstrukturen auf Texturen ab und werden daher zumeist für die Visualisierung von Strömungen in Schichtbildern oder auf Oberflächen verwendet [PVH⁺03]. Van Wijk verwendeten einen texturbasierten Ansatz, bei dem jeder Vektor eines zweidimensionalen Vektorfeldes in einen Konvolutionsfilter umgewandelt und auf eine Textur mit weißem Rauschen angewendet wird, wodurch ein Verschmierungseffekt entsteht [Van91]. Die Funktionsweise dieses Verfahrens wird schematisch links in Abbildung 3.1 dargestellt. Das Ergebnis einer Anwendung des Verfahrens ist in der Mitte zu sehen.

Ein grundlegendes Problem dieses Verfahrens ist die Annahme, dass sich beliebiger Fluss durch eine gerade Flusslinie approximieren lässt. Dies trifft jedoch nur für

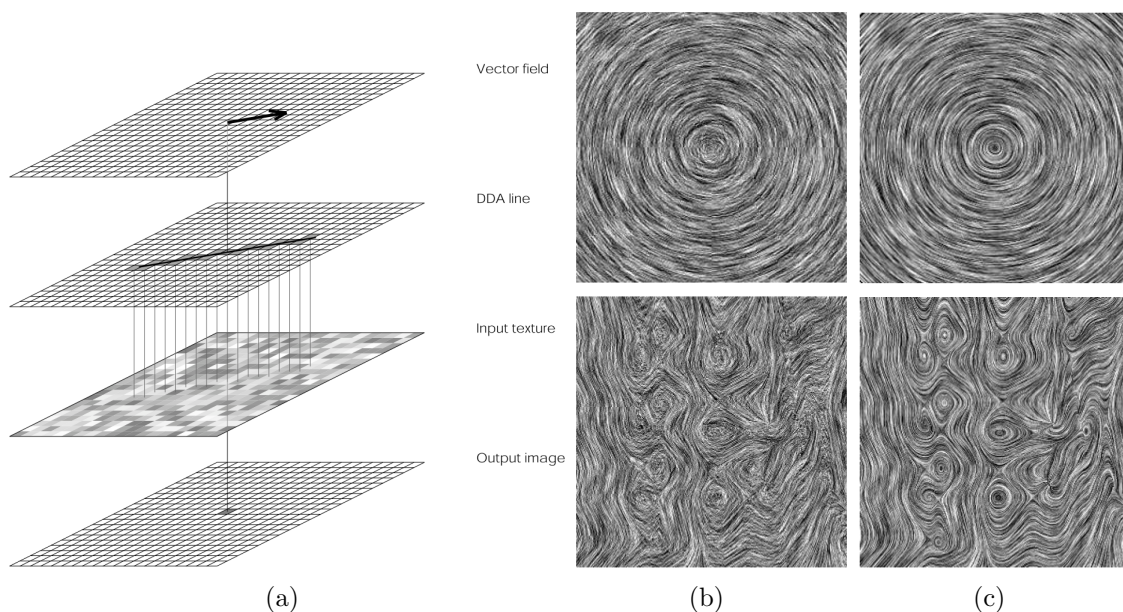


Abbildung 3.1: Funktionsweise des Konvolutionsverfahrens nach Van Wijk (a), Anwendung des Verfahrens auf einen circulären und einen turbulenten Fluss (b), Anwendung des verfeinerten Verfahrens von Cabral und Leedom (c). Quelle: [CL93]

Fluss mit geringen Krümmungsradien zu. Bei komplexen Flussstrukturen, deren Größe die Länge der einzelnen Flusslinien unterschreitet, führt das Verfahren zu einer ungleichmäßigen Visualisierung [CL93]. Ein Beispiel für eine Flussstruktur, die ungenau dargestellt wird, findet sich unten in [Abbildung 3.1 \(b\)](#).

Aus diesem Grund entwickelten Cabral und Leedom mit *Line Integral Convolution (LIC)* eine Erweiterung dieses Verfahrens [CL93]. Anstatt gerade Linien als Konvolutionskernel einzusetzen, wird bei LIC für jeden Pixel eine tangential zum Vektorfeld verlaufende Pfadlinie integriert und als Kernel verwendet. Dies stellt sicher, dass auch kleine und stark gekrümmte Strukturen dargestellt werden können. Der Unterschied beider Verfahren ist in [Abbildung 3.1](#) zwischen den unteren Bildern in (b) und (c) zu erkennen.

Für eine Visualisierung von Flussverläufen in 3D-Vektorfeldern existieren zusätzliche Herausforderungen, wie Verdeckung und räumliche Orientierung [MTHG03]. Hier kommen häufig geometriebasierte Verfahren zum Einsatz, bei denen der Fluss durch geometrische Objekte abstrahiert und dargestellt wird [PVH+03]. Ein Beispiel dafür sind *Integrallinien*, die aus einer Serie von tangential zu den darunter liegenden Vektoren verlaufenden Linien bestehen und eine qualitative Analyse von 3D-Vektorfeldern erlauben [Köh12].

Einen solchen Ansatz stellen Van Pelt et al. vor [vOB+10b]. Zur Ermittlung der Pfadlinien werden Partikel innerhalb einer Seeding-Fläche erzeugt und dann durch das Vektorfeld integriert. Die Pfadlinien geben den Verlauf der einzelnen Partikel wieder und können durch ihre Farbgebung Features, wie beispielsweise Flussgeschwindigkeit, hervorheben.

Da sich die Linien in einer räumlichen Darstellung leicht überschneiden oder gegenseitig verdecken können, ist es hilfreich, Beleuchtungseffekte hinzuzufügen, um die Tiefenwahrnehmung zu verbessern. Ein naiver Ansatz zur Anwendung von Beleuchtung auf Liniensegmente wäre, diese mithilfe dreidimensionaler Geometrie (z.B. als Zylinder) darzustellen. Damit würden sie über eine Oberfläche verfügen, auf die klassische Beleuchtungsmodelle wie beispielsweise *Phong* angewendet werden können. Dies würde jedoch die Erzeugung einer großen Zahl von Grafikprimitiven für die Liniendarstellung erfordern und damit die Anzahl der gleichzeitig in Echtzeit darstellbaren Linien stark verringern.

Um dieses Problem zu umgehen, führen Zöckler et al. *Illuminated Streamlines* ein [ZSH96]. Da nicht-polygonale Integrallinien theoretisch über unendlich viele Normalen verfügen, ist eine Berechnung von Beleuchtung nicht möglich, ohne eine eindeutige Normale für jedes Liniensegment zu definieren. Zöckler et al. verwenden diejenige Normale, die in der von Verlaufsrichtung $\vec{T}$ und Lichtrichtung $\vec{L}$ aufgespannten Ebene liegt. Um die Darstellung zusätzlich zu verbessern, werden schwarze Umrandungslinien (*Halos*) hinzugefügt. Diese erlauben eine verbesserte Tiefenwahrnehmung, da durch die Umrandung eindeutig ersichtlich ist, in welcher Reihenfolge sich Linien gegenseitig verdecken. Ein Beispiel für Illuminated Streamlines ist in [Abbildung 3.2](#) dargestellt.

Born et al. nutzten mit Line Predicates einen Ansatz, um auch komplexe Flüsse untersuchen und deren Darstellung an die eigenen Bedürfnisse anzupassen [BPM+13].

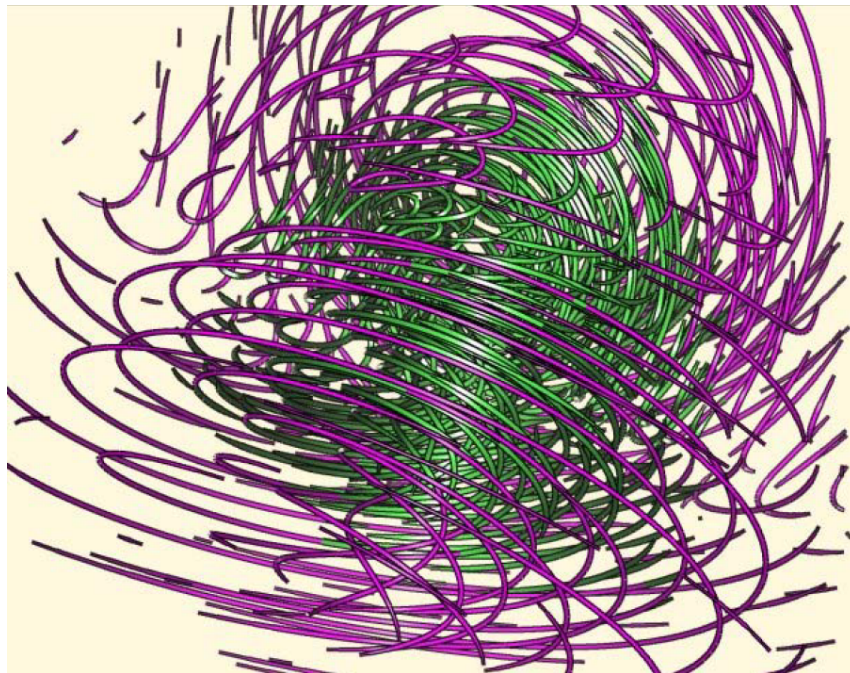


Abbildung 3.2: Mehrfarbige Illuminated Streamlines mit Halos [MTHG03].

Dazu werden Pfadlinien je nach ihren Eigenschaften in Gruppen eingeteilt, wobei eine Pfadlinie simultan Mitglied mehrerer Gruppen sein kann. So können beispielsweise alle Linien gruppiert werden, die eine bestimmte Gefäßregion passieren oder deren Durchschnittsgeschwindigkeit sich in einem definierten Intervall befindet. Durch Ein- und Ausblenden einzelner Gruppen können somit sehr spezifische Flüsse visualisiert werden.

Hummel et al. verwenden Oberflächen anstelle von Linien, um Flussverläufe wiederzugeben [HGH⁺10]. Diese werden aus einzelnen Pfadlinien zusammengesetzt, die wiederum aus einer eindimensionalen Seeding-Linie entspringen. Um auch komplexe Flüsse abbilden zu können, müssen mehrere solcher Oberflächen erzeugt werden. Diese werden daraufhin mit verschiedenen illustrativen Techniken visualisiert, beispielsweise mit Silhouetten, Linientexturen zur Verdeutlichung der Flussrichtung und blickwinkelabhängiger Oberflächentransparenz. Diese erlaubt auch Einblicke in innenliegende Wirbelstrukturen, ohne dass dabei außenliegende Strukturen komplett ausgeblendet werden müssen.

Während bei textur- und geometriebasierten Methoden der gesamte Fluss visualisiert wird, beschränken sich feature-basierte Verfahren auf die Visualisierung einzelner Features innerhalb des Flusses [PVH⁺03]. So führen Stalder et al. zusätzlich zur Visualisierung von Pfadlinien auch eine Wirbelkernextraktion durch [SFH⁺10]. Die gefundenen Wirbelkerne (lila) können dann wie in [Abbildung 3.3](#) dargestellt entweder alleine oder zusätzlich zu den Pfadlinien angezeigt werden.

Zusätzlich zu den vier von Post et al. formulierten Kategorien für Flussvisualisierungen führt Salzbrunn et al. noch die Kategorie der partitionsbasierten Techniken ein. Partitionsbasierte Verfahren gruppieren Gebiete mit ähnlichen Flusscharakteristika und verwenden diese Gruppierungen für die Visualisierung [SJWS08]. Ein Beispiel

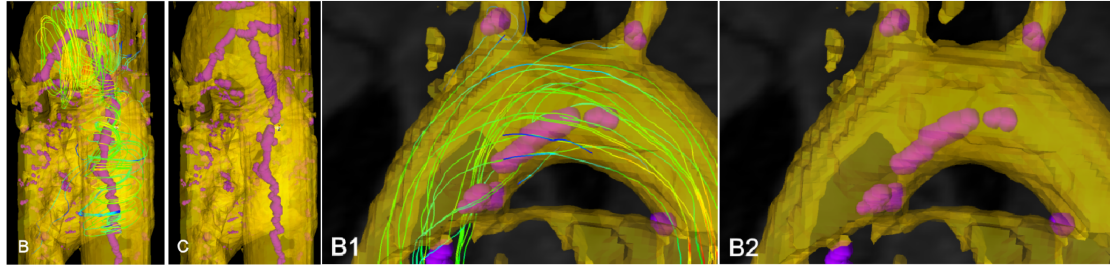


Abbildung 3.3: Visualisierung von Wirbelkernen allein stehend oder zusammen mit Pfadlinien nach Stalder et al. [SFH⁺10].

hierfür liefern Heckel et al., die ein Clustering des Vektorfeldes anhand der Richtung und Länge der Vektoren vornehmen [HWHJ99]. Zu Beginn des Algorithmus existiert nur ein einziges Cluster, welches so lange weiter aufgeteilt wird, bis eine Abbruchbedingung erreicht wird. Heckel et al. verwenden hierbei die maximale Abweichung der Vektoren vom Durchschnittsvektor eines Clusters. Fällt dieser unter einen bestimmten Wert, wird das Cluster nicht weiter aufgeteilt. Nach Abschluss des Clusterings kann jedes Cluster durch dessen Durchschnittsvektor repräsentiert und visualisiert werden.

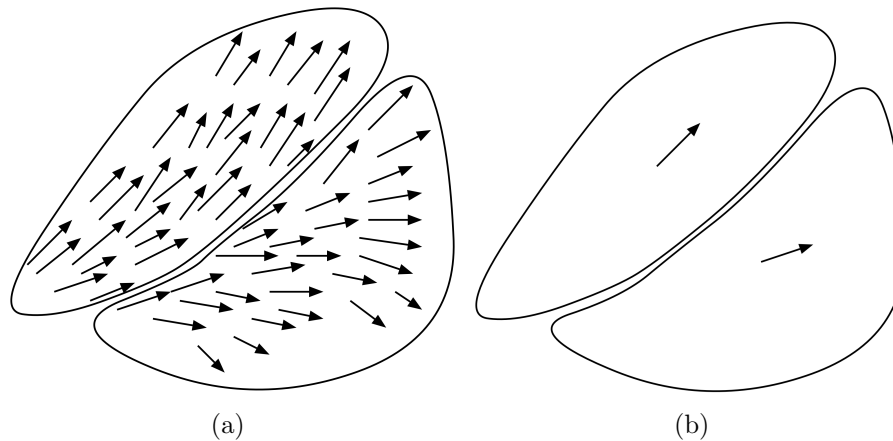


Abbildung 3.4: Clustering eines Vektorfeldes (a) sowie Repräsentation jedes Clusters durch einen einzelnen Vektor (b). Quelle: [HWHJ99]

3.3 Wirbelextraktion

Die in Abschnitt 3.1 beschriebenen Studien stützen sich größtenteils auf die manuelle Beobachtung animierter Pfadlinien. Dies birgt jedoch Risiken, da Wirbel leicht übersehen oder deren Größe falsch eingeschätzt werden können. Ein Wirbel lässt sich als eine kreis- oder spiralförmige Bewegung innerhalb eines Fluids bezeichnen [HPT⁺08]. Um Wirbel detektieren und quantifizieren zu können, wird jedoch eine mathematische Beschreibung benötigt, die für eine beliebige Struktur festlegt, inwieweit diese einen Wirbel darstellt. Eine einheitliche mathematische Definition eines Wirbels existiert jedoch nicht.

Bei Wirbelextraktionsmethoden kann grundsätzlich zwischen linienbasierten und regionenbasierten Ansätzen unterschieden werden. Linienbasierte Ansätze identifizie-

ren Wirbelkerne und repräsentieren diese als sog. *Wirbelkernlinien*. Regionenbasierte Ansätze hingegen identifizieren Regionen, in denen Verwirbelungen auftreten. Zusätzlich ist auch eine Unterteilung in lokale oder globale Verfahren möglich [HJ05]. Bei lokalen Verfahren wird lediglich die unmittelbare Nachbarschaft eines Punktes untersucht, um festzustellen, ob dieser Teil einer Wirbelstruktur ist. Globale Verfahren hingegen beziehen auch weiter entfernte Bereiche des Vektorfeldes in die Berechnung ein.

Eine relativ einfache, jedoch fehlerbehaftete Methode zur Wirbelextraktion ist die Untersuchung der Rotation an jedem Punkt im Vektorfeld. Dafür wird der *Curl-Vektor* verwendet, der angibt, um welche Achse sich die Strömung an einem bestimmten Punkt dreht:

$$\mathbb{V}(p) = \begin{pmatrix} u \\ v \\ w \end{pmatrix} \quad \text{curl}(\mathbb{V}(p)) = \begin{pmatrix} w_y - v_z \\ u_z - w_x \\ v_x - u_y \end{pmatrix} \quad (3.1)$$

Die Länge des Curl-Vektors und damit die Stärke der Rotation wird als *Vorticity* bezeichnet. Geht man davon aus, dass sich im Kern eines Wirbels hohe Vorticity-Werte ermitteln lassen, so bietet sich ein Schwellenwertverfahren an, um Wirbelkerne zu detektieren. Hierbei handelt es sich um eine lokale, linienbasierte Methode. Problematisch bei diesem Verfahren ist zum einen, dass dieser Schwellenwert nicht berechnet [Kha11], sondern nur anhand von Histogrammanalysen geschätzt werden kann. Zudem können hohe Vorticity-Werte auch in Bereichen auftreten, in denen sich keine Wirbel befinden. [Abbildung 3.5](#) zeigt dieses Phänomen exemplarisch an einem Rohr mit laminarem (gradlinigem) Fluss. Gleichzeitig gibt es Wirbelstrukturen, die keine erhöhten Vorticity-Werte verursachen.

Levy und Degani verwendeten die sog. *Helicity* H_n , um den Kern eines Wirbels zu finden [LDS90]. Dabei wird von der Annahme ausgegangen, dass Geschwindigkeit $\vec{v}$ und Curl-Vektor $\vec{w}$ im Kern eines Wirbels die gleiche Richtung haben und der Betrag des Cosinus zwischen beiden demzufolge den Wert 1 annimmt.

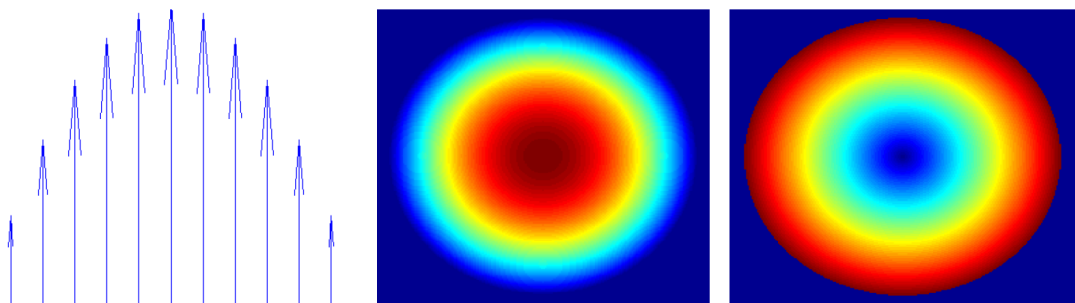


Abbildung 3.5: Falscherkennung von Wirbeln in einem Rohr mit laminarem Fluss; Parabolisches Geschwindigkeitsprofil (links) und Ebene mit Geschwindigkeiten (Mitte), Stärke der Vorticity (rechts)) [Kha11]

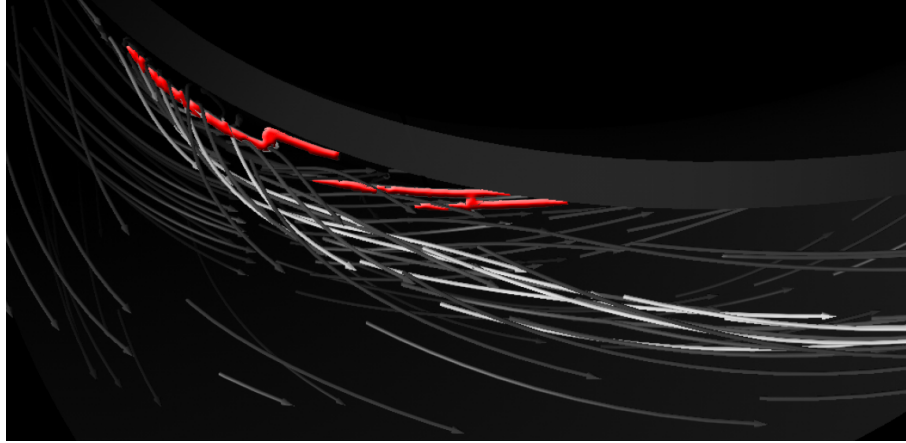


Abbildung 3.6: Fehlerhaft erkannter Wirbelkern (rot) in helikalem Fluss [Rot00].

Damit ergibt sich folgende Formel für die Helicity H_n :

$$H_n = \frac{\vec{v} \cdot \vec{w}}{|\vec{v}| \cdot |\vec{w}|} \quad (3.2)$$

Punkte, an denen $|H_n| = 1$ gilt, liegen im Zentrum eines Wirbels, dessen Drehrichtung sich am Vorzeichen von H_n ablesen lässt. Um die Kernlinie eines Wirbels zu verfolgen, werden ausgehend von Punkten mit maximaler Helicity Pfadlinien integriert, die wie in [Abbildung 3.6](#) dargestellt jedoch nicht zwangsweise den korrekten Verlauf des Wirbels wiedergeben. Auch dieses Verfahren arbeitet lokal und linienbasiert.

Einen ähnlichen Ansatz verfolgen Banks und Singer mit ihrer *Prediction-Corrector* Methode [BS95]. Mithilfe von Schwellenwerten versuchten sie, Punkte zu identifizieren, die sich durch besonders hohe Vorticity und gleichzeitig niedrigen Druck auszeichneten, da sich derartige Punkte mit hoher Wahrscheinlichkeit innerhalb eines Wirbelkerns befinden. Ausgehend von diesen Punkten wird der Rest des Wirbelkerns schrittweise rekonstruiert. Der jeweils nächste Punkt wird zunächst durch Integration des Curl-Vektors ermittelt. Banks und Singer beobachteten jedoch, dass dieser meist nicht exakt auf der Wirbelkernlinie liegt, ihr jedoch recht nahe kommt. Um diesen Fehler zu korrigieren, wird in den Punkt eine Ebene gelegt, die senkrecht auf dem Curl-Vektor steht. Innerhalb dieser Ebene wird nach einem lokalen Druck-Minimum gesucht und der Punkt an dieses Druckminimum verschoben. Dieser Vorgang wird in [Abbildung 3.7](#) verdeutlicht. Da diese Methode nicht nur die unmittelbare Nachbarschaft eines Punktes untersucht, handelt es sich hierbei um ein globales linienbasiertes Verfahren.

Auch Jeong und Hussain stützen sich in ihrem Ansatz auf die Annahme, dass im Wirbelkern ein lokales Druckminimum auftritt [JH94]. Dieses stellt allerdings weder eine ausreichende noch notwendige Bedingung für das Vorhandensein eines Wirbels dar, denn auch andere Strömungsstrukturen können Druckminima hervorbringen und es existieren Wirbelstrukturen, die kein Druckminimum ausbilden. Mit dem λ_2 -Wirbelkriterium versuchen Jeong und Hussain, diese Probleme zu umgehen. Dazu

bilden sie die Jacobi-Matrix J des Vektorfeldes und spalten diese in ihren symmetrischen Deformations- oder Scherungsteil S und ihren asymmetrischen Rotationsteil Ω auf.

$$S = \frac{J + J^T}{2} \quad \Omega = \frac{J - J^T}{2} \quad (3.3)$$

Ein Wirbel wird nun als eine verbundene Region definiert, in der die reelle und symmetrische Matrix $S^2 + \Omega^2$ über zwei negative Eigenwerte verfügt. Sind λ_1, λ_2 und λ_3 die Eigenwerte dieser Matrix und gilt $\lambda_1 \geq \lambda_2 \geq \lambda_3$, so existiert dort ein Wirbel, wo der zweite Eigenwert λ_2 kleiner als 0 ist. Im Test mit verschiedenen Flussmustern zeigen Jeong und Hussain, dass ihre λ_2 -Methode im Vergleich zu anderen Verfahren, wie der Suche nach Vorticity-Maxima, bessere Ergebnisse erzielt [JH94].

Die Wirbelerkennung mittels λ_2 stellt ein lokales, regionenbasiertes Verfahren zur Wirbelextraktion dar und wird aufgrund ihrer Effektivität häufig angewendet [Rot00] [BMDH07]. Ein Beispiel ist die Arbeit von Köhler [Köh12]. Hier wird die Wirbelextraktion mittels λ_2 genutzt, um Verwirbelungen in mithilfe von 4D PC-MRI gemessenen Blutflussdaten zu finden und hervorzuheben. Für die Visualisierung sowie die Begrenzung der Pfadlinien auf das Gefäß ist eine manuelle Segmentierung erforderlich. Daraufhin werden im Gefäß Pfadlinien berechnet, die den Blutfluss wiedergeben. Der Anwender kann diese anhand ihrer λ_2 -Werte filtern und somit laminaren Fluss ausblenden. Ein Beispiel dieser Visualisierung wird in [Abbildung 3.8](#) dargestellt.

Zudem führt Köhler in seiner Arbeit einen Vergleich verschiedener Wirbelerkennungsverfahren durch. Als einen der Favoriten gibt er das λ_2 -Verfahren an. Positive Erwähnung findet, dass dieses robust gegenüber Rauschen ist und Verwirbelungen weitgehend zuverlässig erkennt.

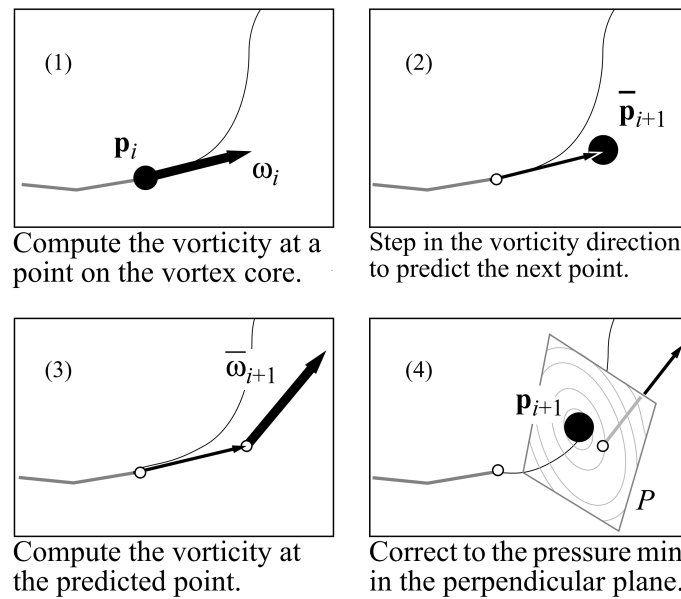


Abbildung 3.7: Schrittweise Verfolgung der Wirbelkernlinie mittels *Prediction-Corrector* [BS95].

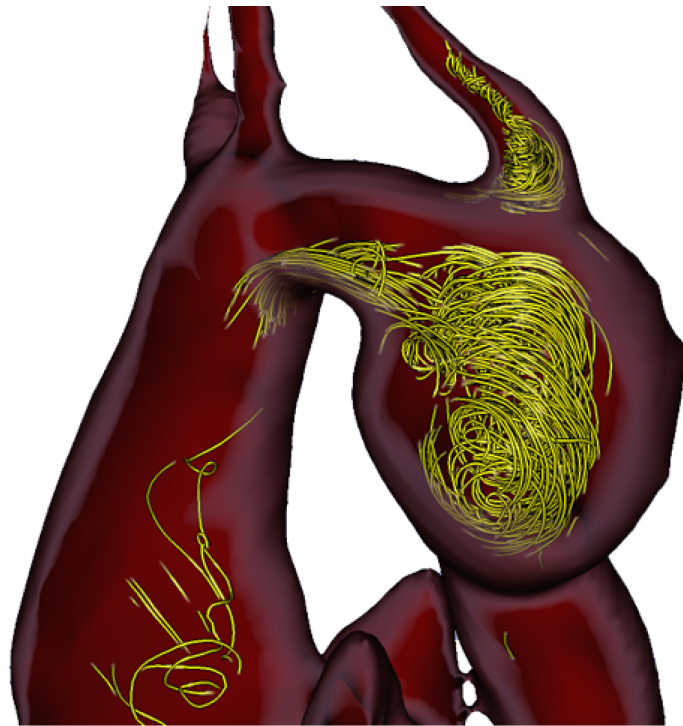


Abbildung 3.8: Visualisierung einer Aorta mit enthaltenen Wirbeln nach Köhler [Köh12].

4. Vollautomatische Visualisierung von kardialen 4D PC-MRI Blutflussdaten mit intelligenter Fokussierung auf Features

Bevor die Visualisierung von Anatomie und Flussdaten durchgeführt werden kann, müssen die Eingangsdaten aufbereitet werden. Diese liegen in Form von zweidimensionalen DICOM-Dateien vor, wobei pro Schicht und Zeitschritt je eine Datei existiert. Zunächst werden die vorhandenen Dateien eingelesen und im Speicher der Grafikkarte in Form dreidimensionaler Texturen abgelegt. Da die Grafikkarte eine Verwendung vierdimensionaler Texturen nicht vorsieht, wird jeder Zeitschritt in einer eigenen 3D-Textur gespeichert. Zudem werden verschiedene Informationen vorberechnet, um die Echtzeitvisualisierung zu entlasten. Da die Daten klein genug sind, um vollständig im Speicher der Grafikkarte abgelegt zu werden, ist Streaming nicht erforderlich.

Die Visualisierung der Anatomie wird durch direktes Volumenrendering auf Basis der tMIP-Daten durchgeführt. Die Pfadlinien hingegen werden durch die Verwendung von Grafikprimitiven gerendert und basieren auf dem aus den Flussdaten generierten Vektorfeld. Erzeugt werden die Pfadlinien durch Partikelintegration. Die Erkennung von Verwirbelungen erfolgt mithilfe des λ_2 -Wirbelkriteriums.

4.1 DVR

Direct Volume Rendering ist für eine vollautomatische Visualisierung besser geeignet als indirekte Visualisierung. Letztere kann nur dann vollautomatisch durchgeführt werden, wenn sich ein geeigneter Iso-Wert für die Darstellung ermitteln lässt. Dies ist jedoch aufgrund des geringen Kontrastes der Ausgangsdatensätze sowie des Fehlens von eindeutigen Zuordnungen zwischen Gewebetypen und Intensitätswerten nicht möglich. Zudem ist die manuelle Anpassung eines fehlerhaft ermittelten Iso-Wertes wesentlich zeitaufwendiger.

4.1.1 Rendering-Pipeline

Das direkte Volumen-Rendering zur Darstellung der Anatomie wird in mehreren Schritten durchgeführt, die in der folgenden Pipeline beschrieben werden. Zur Anatomievisualisierung wird eine tMIP der Magnitude-Bilder verwendet.

4.1.1.1 Clearing und Farbwürfel

Da der Speicher der Grafikkarte über mehrere Rendering-Vorgänge persistent ist, müssen vor dem Beginn der eigentlichen Berechnungen alle angefallenen Daten vorhergehender Visualisierungsvorgänge in einem Clearing-Schritt entfernt werden.

Der Großteil dieser Daten liegt in Form von *Shader Storage Buffer Objects* (SSBOs) vor, in denen je ein oder mehrere Werte pro Pixel gespeichert sind. Daher bietet es sich an, die Löschung dieser Objekte im Fragment Shader durchzuführen. Zu diesem Zweck wird ein bildschirmfüllendes Rechteck (siehe [Abschnitt 2.3.1.2](#)) mit einem gebundenen Clearing-Shader gezeichnet, der die Buffer mit 0 als Standardwert füllt.

Anschließend wird ein Farbwürfel erzeugt, dessen Vorder- und Rückseite separat gerendert und gespeichert werden. Da der Datensatz dieselben Dimensionen auf dem Bildschirm einnehmen wird wie der Farbwürfel, muss skaliert und an die Größenverhältnisse des Datensatzes angepasst werden.

Dazu wird ein Einheitswürfel zwischen den Positionen $(0, 0, 0)$ und $(1, 1, 1)$ gezeichnet, der zunächst entsprechend der Größenverhältnisse des Datensatzes skaliert und danach je nach Einstellung des Nutzers in der Benutzeroberfläche rotiert bzw. positioniert wird. Zum Skalieren des Würfels wird ein Skalierungsvektor $\vec{S}$ aus dem Dimensionsvektor $\vec{D}$ des Datensatzes (bestehend aus dessen Breite w , Höhe h und Tiefe d) gebildet, der durch dessen längste Dimension geteilt wird:

$$\vec{D} = \begin{pmatrix} w \\ h \\ d \end{pmatrix} \qquad \vec{S} = \frac{\vec{D}}{\max(w, h, d)} \quad (4.1)$$

Höhe, Breite und Tiefe werden wiederum aus der Voxelgröße sowie der Anzahl der Gitterpunkte in der entsprechenden Richtung gebildet. Durch eine Kombination aus Vertex- und Fragment-Shader wird jeder Punkt auf der Oberfläche entsprechend seiner Position im Koordinatensystem eingefärbt. Der Vertex Shader erzeugt dabei die Farbwerte pro Eckpunkt, der Fragment Shader interpoliert diese für die dazwischen liegenden Flächen.

An dieser Stelle wird auch das erste Mal die Notwendigkeit des Clearing-Schrittes deutlich. Da der Fragment Shader in diesem Schritt nur für diejenigen Pixel aufgerufen wird, auf die der Würfel im aktuellen Rendering-Vorgang abgebildet wird, bleiben möglicherweise Werte aus vorhergehenden Rendering-Vorgängen erhalten, die Bildfehler verursachen würden.

4.1.1.2 Volume Rendering

Mithilfe der Farbwerte von Vorder- und Rückseite des Würfels werden Strahlen durch den Volumendatensatz gesendet. Grundsätzlich gibt es zwei Möglichkeiten, dessen Abtastrate anzugeben. Die erste ist, für jeden Strahl die gleiche Anzahl an Samples zu verwenden. Dies führt allerdings dazu, dass bei vergleichsweise kurzen Strahlen, die beispielsweise nur eine Ecke des Volumens schneiden, auf wenig Raum sehr viele Sample-Punkte gesetzt werden. Kombiniert man dies mit der Annahme, dass die Anatomie eher im Zentrum des Datensatzes positioniert ist, werden hier viele Sample-Punkte unnötig berechnet.

Die andere Möglichkeit ist, einen konstanten Abstand zwischen Sample-Punkten zu definieren. Auf diese Weise enthalten kürzere Strahlen weniger Punkte, was zu einer Verbesserung der Verarbeitungsgeschwindigkeit führt. Zudem lässt sich ein konstanter Abstand besser an die Gegebenheiten des Datensatzes, beispielsweise dessen räumlicher Auflösung, anpassen. Dem Nutzer stehen dazu mehrere Voreinstellungen

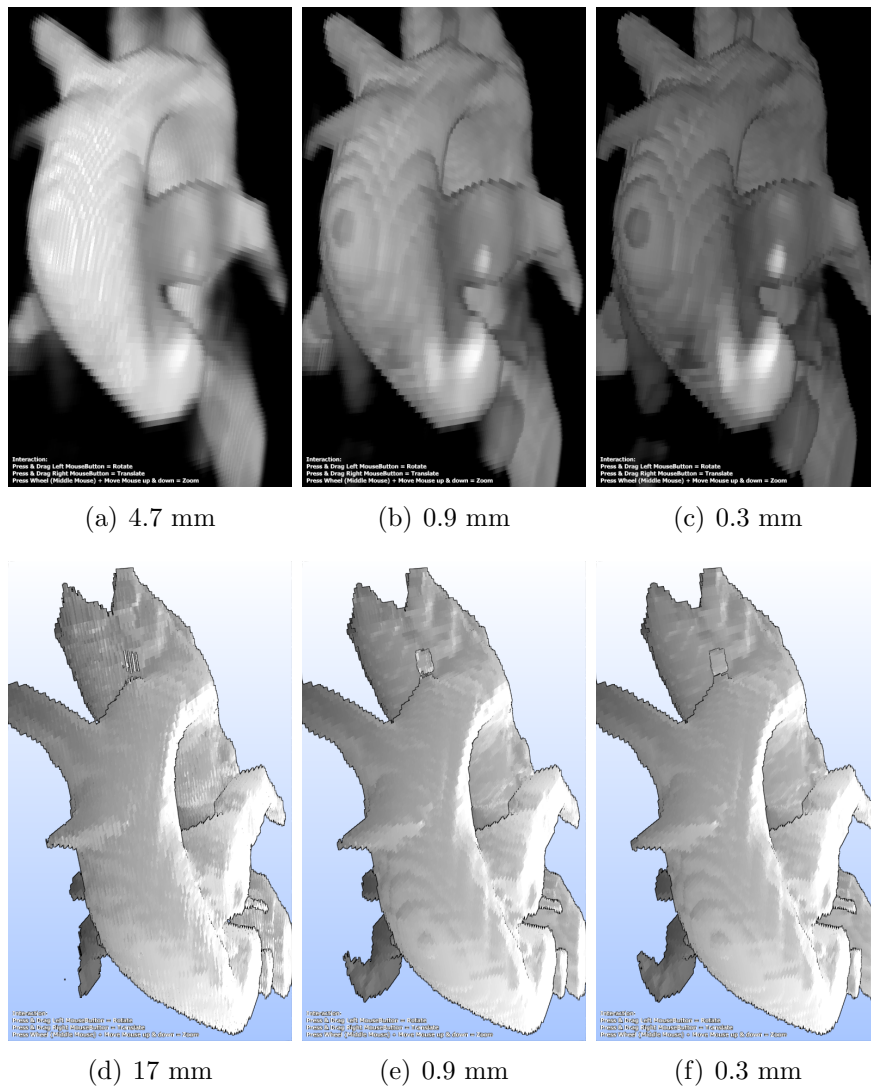


Abbildung 4.1: Einfluss der verschiedenen vordefinierten Schrittweiten auf unterschiedliche Visualisierungsformen.

im Bereich zwischen 4.7 mm und 0.3 mm zur Verfügung, wobei eine Unterschreitung von 0.9 mm kaum optische Veränderungen mit sich bringt. Dies wird auch in [Abbildung 4.1](#) deutlich. Während die Visualisierungen bei hoher Sample-Distanz verstärkt verschiedene Störungen wie Aliasing-Artefakte enthalten, sind diese bei mittlerer Distanz kaum noch sichtbar. Die Darstellungen mit geringer Sample-Distanz lassen sich kaum von denen mit mittlerer Distanz unterscheiden, benötigen gleichzeitig jedoch wesentlich mehr Zeit zum Rendern.

Die Visualisierung des Datensatzes findet vollständig im Fragment Shader statt, der durch das bildschirmfüllende Rechteck für jeden Bildpunkt genau einmal aufgerufen wird. Der Standardwert für diese Einstellung liegt bei 0.9 mm, sodass Datensätze mit einer Voxelbreite von 1.8 mm und höher ohne das Auftreten von Aliasing-Artefakten visualisiert werden können.

Der Shader iteriert über jeden Sample-Punkt auf dem Strahl und ermittelt die entsprechenden Datenwerte mithilfe trilinearer Interpolation, welche auf der GPU für Texturen hardwareseitig unterstützt wird. Die Abbruchbedingung für diese Iteration ist das Verlassen des Volumens sowie weitere, von der Darstellungsart abhängige Faktoren.

4.1.2 Darstellung der Anatomie

Insgesamt wurden drei unterschiedliche Darstellungsarten implementiert, die sich jedoch alle auf die aus dem Strahl extrahierten Datenwerte stützen. Zwischen diesen Darstellungsarten kann jederzeit gewechselt werden.

4.1.2.1 MIP

Analog zur Beschreibung in [Abschnitt 2.3.3](#) steht eine MIP-Darstellung zur Verfügung. Dabei wird der größte im Datensatz vorhandene Intensitätswert auf Weiß, der niedrigste auf Schwarz abgebildet; dazwischen liegende Werte werden interpoliert. Tiefeninformationen gehen bei dieser Darstellungsart verloren.

4.1.2.2 Erweiterte Rampenfunktion

Beim Einsatz einer Rampenfunktion als Transferfunktion hingegen werden Tiefeninformationen erhalten. Eine klassische Rampenfunktion wird entweder durch ihren Mittelpunkt (*center*) sowie ihre Breite (*width*) oder durch einen Start- und Endwert definiert. [Abbildung 4.2](#) zeigt auf der linken Seite ein Beispiel für eine solche Rampenfunktion.

Mathematisch wird eine Rampenfunktion üblicherweise folgendermaßen definiert:

$$\begin{aligned}
 \text{ramp} : \mathbb{R} &\rightarrow [0, 1]^2 \\
 \text{ramp}(x) &= \begin{cases} (0, 0) & \text{für } x \leq b_1 \\ (1, 1) & \text{für } x \geq b_2 \\ (r(x), r(x)) & \text{sonst} \end{cases} \quad \text{mit } b_1 < b_2 \\
 r(x) &= \frac{(x - b_1)}{(b_2 - b_1)}
 \end{aligned} \tag{4.2}$$

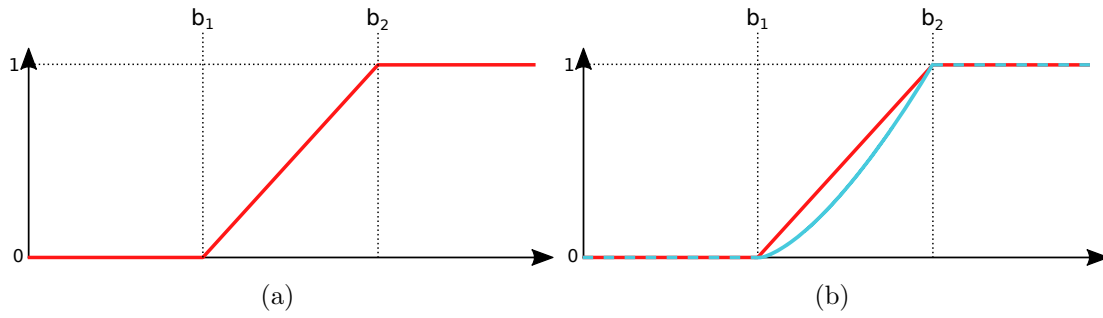


Abbildung 4.2: Schematische Darstellung einer üblichen Rampenfunktion (a), sowie der verwendeten erweiterten Rampenfunktion mit Opazität (rot) und Grauwert (blau) (b), beide gekennzeichnet durch den Beginn der Rampe (b_1) sowie deren Ende (b_2).

In dieser Definition wird derselbe Wert für Opazität und Grauwert verwendet. Dies hat den Nachteil, dass ein vergleichsweise großer Teil des Grauwertbereichs für Voxel verwendet wird, die durch ihre geringe Opazität ohnehin kaum sichtbar sind. Eine Lösung für dieses Problem stellt die Verwendung unterschiedlicher Rampenfunktionen für Opazität und Grauwert dar, allerdings erfordert dies vom Nutzer die Eingabe zweier zusätzlicher Schwellenwerte.

Um dies zu vermeiden, können statt unterschiedlichen Schwellenwerten auch nur unterschiedliche r -Funktionen verwendet werden:

$$ramp'(x) = \begin{cases} (0, 0) & \text{wenn } x \leq b_1 \\ (1, 1) & \text{wenn } x \geq b_2 \text{ mit } b_1 < b_2 \\ (r_g(x), r(x)) & \text{sonst} \end{cases} \quad (4.3)$$

$$r_g(x) = r(x)^{1.5}$$

Die Opazität wird nach wie vor linear interpoliert, wie rechts in [Abbildung 4.2](#) zu erkennen, steht jedoch für höhere Opazitäten ein größerer Grauwertbereich zur Verfügung. Der Einfluss des Exponenten auf die Visualisierung wird in [Abbildung 4.3](#) gezeigt.

Die Transferfunktion wird vom Anwender über die Benutzeroberfläche eingestellt. Der Start- bzw. Endpunkt der Rampe wird beliebig innerhalb des Wertebereichs $[0, 1]$ platziert, wobei der Endpunkt größer oder gleich dem Startpunkt sein muss. Die Funktionsweise der Rampenfunktion wird in [Abschnitt 2.3.4](#) beschrieben.

Erreicht die akkumulierte Opazität den Wert 0.95, wird die Berechnung nach dem Prinzip der *Early Ray Termination* (siehe [Abschnitt 2.3.3](#)) abgebrochen. Kreuzt der Strahl einen Partikelpfad, so wird bei jedem Sample-Punkt zusätzlich mittels eines Vergleichs der Tiefenwerte geprüft, ob der Punkt sich hinter dem Pfad befindet. Ist dies der Fall, so wird die Iteration des Strahls vorzeitig gestoppt, da Partikelpfade vollständig opak sind.

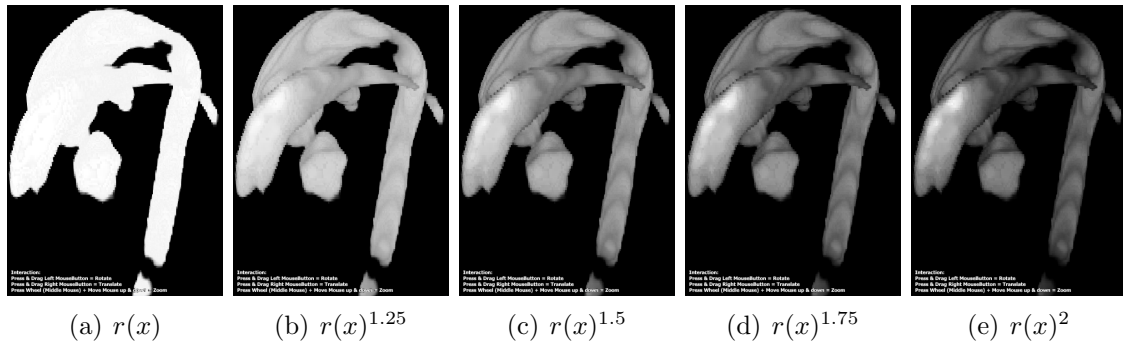


Abbildung 4.3: Vergleich unterschiedlicher Exponenten für den Grauwert-Teil der Rampenfunktion.

4.1.2.3 Solider Körper

Diese Visualisierung arbeitet ähnlich wie die MIP ohne Opazitätswerte, allerdings bleiben Tiefeninformationen erhalten, indem anstelle einer Transferfunktion ein einzelner Schwellenwert zum Einsatz kommt. Sobald die Intensität eines Punktes s_p diesen Schwellenwert überschreitet, wird der entsprechende Pixel eingefärbt. Wird der Schwellenwert entlang des gesamten Strahls nicht überschritten, bleibt der Pixel transparent. Dadurch entsteht eine explizite Oberfläche, die Gefäß und Umgebung scharf trennt. Zusätzlich zum Grauwert wird der Tiefenwert sowie der Gradient von s_p in einem Storage Buffer gespeichert. Diese Informationen werden für die Nachbearbeitung der Visualisierung verwendet, bei der zusätzliche optische Effekte erzeugt werden.

Der Grauwert jedes Pixels wird basierend auf der Entfernung des zugrunde liegenden Punktes s_p vom Betrachter bestimmt. Weiter entfernte Teile der Anatomie werden somit dunkler gezeichnet. Mithilfe der Gradienten wird eine Beleuchtung der Oberfläche mittels Phong-Modell durchgeführt, um die Plastizität der Darstellung zu erhöhen. [Abbildung 4.4](#) zeigt deutlich, wie die Verwendung von Beleuchtung die Darstellung der Oberfläche verbessert.

Zusätzlich verstärkt wird dieser Effekt durch das Hinzufügen einer Silhouette um die Anatomie. Hierfür werden die gespeicherten Tiefenwerte verwendet. Im Nachbearbeitungsschritt wird für jeden Wert in diesem Buffer die Summe der Differenz zwischen dem Wert selbst und den 8 Punkten in seiner Umgebung berechnet. Übersteigt diese den Schwellenwert von 0.015, so wurde ein Tiefenübergang gefunden und der entsprechende Pixel im fertigen Bild wird schwarz gefärbt. Auch dies wird in [Abbildung 4.4](#) dargestellt. Der Schwellenwert von 0.015 wurde empirisch bestimmt und entspricht in etwa 5 mm. Dieser Wert stellt sicher, dass die Ränder der Anatomie zuverlässig gefunden, jedoch keine falschen Ränder beispielsweise an Voxelübergängen oder stark geneigten Oberflächen markiert werden.

4.1.3 Automatische Transferfunktionen und Schwellenwerte

Da weder eine feste Zuordnung von Intensitätswerten und Gewebetypen (wie beispielsweise bei der Hounsfield-Skala) noch eine Vorsegmentierung der Daten vorliegt, erfolgt die Generierung der Transferfunktion auf Basis eines Histogramms.

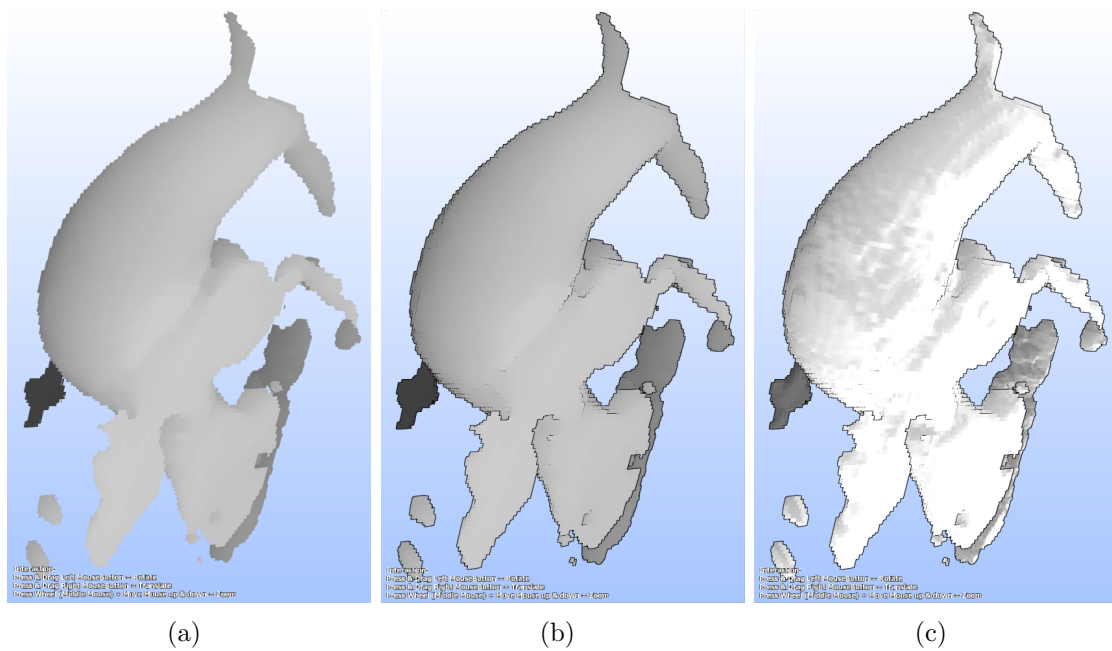


Abbildung 4.4: Vergleich der verschiedenen Nachbearbeitungsschritte bei der soliden Visualisierung: (a) Ohne Nachbearbeitung, (b) mit Rändern sowie (c) mit Rändern und Beleuchtung

Ist die erweiterte Rampenfunktion als Visualisierung eingestellt, so muss der Start- und Endpunkt der Rampe berechnet werden. In empirischen Tests hat sich hier das 40%-Quantil für den Startpunkt sowie das 90%-Quantil für den Endpunkt bewährt. Mit diesen Einstellungen wird die Anatomie kontrastreich dargestellt, während Umgebung und eventuelle Artefakte weitgehend ausgeblendet werden. Bei den soliden Visualisierungen ist mit dem Schwellenwert nur ein einziger Wert automatisch festzulegen. Für diesen hat sich das 50%-Quantil bewährt. Sowohl der Schwellenwert als auch Start- und Endpunkt der Rampenfunktion sind in [Abbildung 4.5](#) zusammen mit dem Histogramm eines Datensatzes dargestellt.

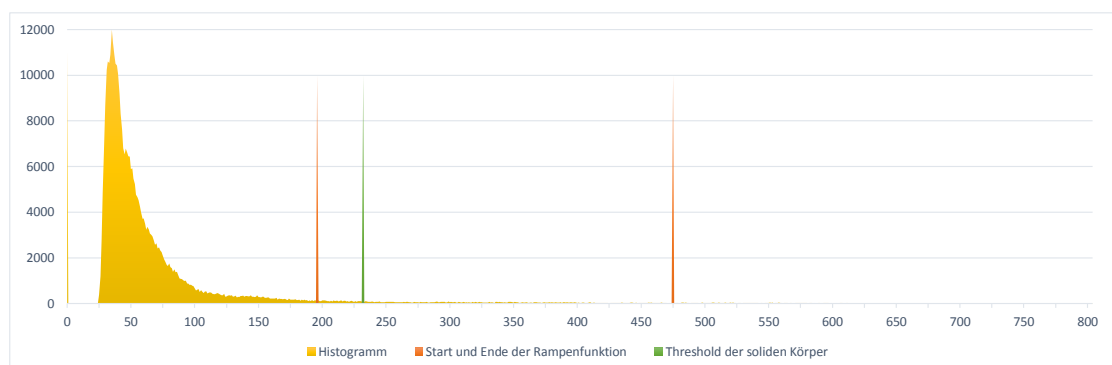


Abbildung 4.5: Histogramm eines Datensatzes (gelb) mit eingezeichneten Quantilen für Rampenfunktion (orange) sowie solide Körper (grün).

4.1.4 Tiefenberechnung

Beim direkten Volumenrendering existiert kein Z-Buffer, da anstelle von Polygonen Strahlen durch das Volumen im Fragment Shader berechnet werden. Zwar kann ein Pseudo-Tiefenwert eines Punktes im Datensatz basierend auf seiner Position auf dem Strahl erzeugt werden. Allerdings zeigt dieser mitunter, wie in [Abbildung 4.6](#) dargestellt, starke Artefakte.

Dieses Problem kann behoben werden, indem beim Rendern des aus Polygonen bestehenden Farbwürfels der Z-Buffer ausgelesen wird. Nach einer Linearisierung enthält dieser die Längen der in [Abbildung 4.6](#) grau dargestellten Linien - durch simple Addition mit dem aus dem Strahl berechneten Tiefenwert (blau) kann die tatsächliche Entfernung wiederhergestellt werden.

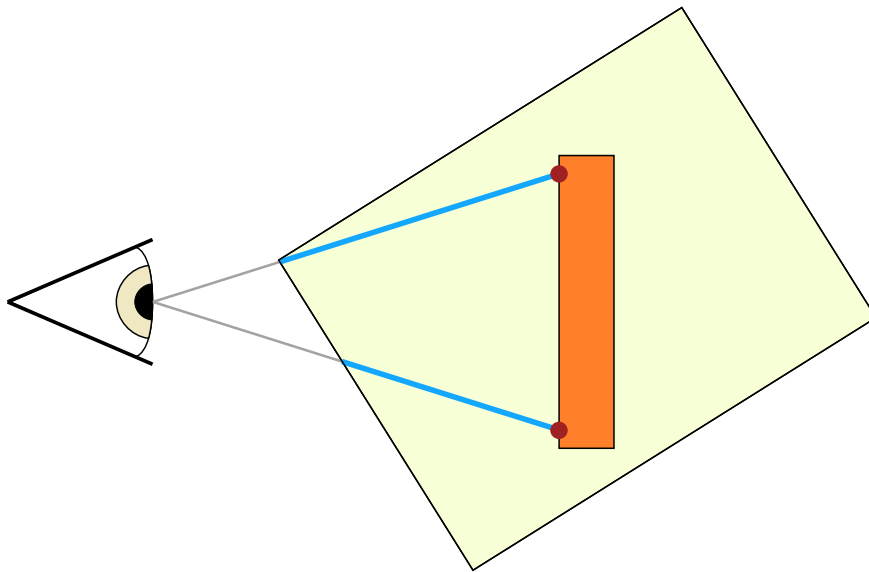


Abbildung 4.6: Tiefenwert-Artefakte. Obwohl beide Punkte (rot) auf der Anatomie (orange) gleich weit vom Betrachter entfernt sind, entstehen aufgrund der Ausrichtung des Volumens (hellgrün) unterschiedliche Tiefenwerte (blau).

4.2 Partikelvisualisierung

Die Verlaufsrichtung der Blutes kann auf unterschiedliche Weise visualisiert werden. Eine Möglichkeit ist die Vorberechnung von Pfadlinien, die während der Animation basierend auf ihrer temporalen Position ein- und ausgeblendet werden. Dies würde den Rechenaufwand während der Echtzeitvisualisierung reduzieren, jedoch auch die Anpassbarkeit der Visualisierung einschränken. Die Berechnung der Pfadlinien müsste bei jeder Anpassung der Anatomievisualisierung vom Grunde auf neu berechnet werden, sodass Änderungen der Visualisierung nicht mehr in Echtzeit durchgeführt werden können.

Als Alternative bietet sich die Verwendung von Partikeln an, welche an zufälligen Positionen innerhalb der Anatomie initialisiert werden und dem Blutfluss folgen. Anstatt den vollständigen Pfad über alle Zeitschritte vorzuberechnen, werden bloß die letzten n Positionen der Partikel gespeichert und als Pfadlinien visualisiert. Bei

Modifikation der Gefäßvisualisierung und einer damit verbundenen Änderung der sichtbaren Anatomie können einzelne Partikel entfernt und entsprechend der neuen Visualisierung positioniert werden. Im Gegensatz zu vorberechneten Pfadlinien ermöglicht dieser Ansatz jedoch keine beliebigen Bewegungen oder Sprünge auf der Zeitlinie - die Animation kann nur vorwärts berechnet werden, ein Sprung zurück zu vorhergehenden Zeitpunkten ist nicht möglich.

4.2.1 Seeding der Partikel

Um sicherzustellen, dass Partikel nur in den Blutgefäßen platziert werden und diese auch durch ihre Bewegung nicht verlassen, wird durch einen Threshold-Filter eine binäre Maske auf den tMip-Daten generiert. Der für den Filter erforderliche Schwellenwert ist an die Einstellungen des DVR gekoppelt, sodass Partikel nur in den Bereichen entstehen können, die aktuell sichtbar sind.

Um eine möglichst vollständige Abdeckung des gesamten Gefäßes zu erreichen und damit alle vorhandenen Strukturen im Fluss zu visualisieren, ist eine ausreichende Anzahl an Partikeln erforderlich. [Abbildung 4.7](#) zeigt die gleiche Partikelvisualisierung mit einer unterschiedlichen Anzahl von Partikeln. In empirischen Tests hat sich gezeigt, dass erst ab einer Zahl von 2048 eine gute Füllung des Gefäßes und damit eine zuverlässige Sichtbarkeit von Features wie Wirbeln in den verwendeten Datensätzen gewährleistet ist.

Viele der Partikel repräsentieren laminaren Fluss, der allerdings keine hohe Partikelzahl für eine exakte Repräsentation erfordert. Befinden sich Verwirbelungen im Inneren des Gefäßes, so können diese durch umliegenden laminaren Fluss sogar verdeckt werden. Um die Zahl dieser Partikel zu reduzieren, hat der Anwender die Option, ein ungleichmäßiges Seeding zu aktivieren. Dadurch wird der Seeding-Prozess in zwei Teile aufgespalten. Von den insgesamt n Partikeln werden $\frac{n}{2}$ gleichmäßig positioniert. Die restlichen Partikel verwenden eine andere Seeding-Maske, die sich innerhalb der normalen binären Maske befindet und die Umgebung aller Bereiche umfasst, in denen ein erhöhter λ_2 -Wert auftritt. Die Umgebung dieser Gebiete kann

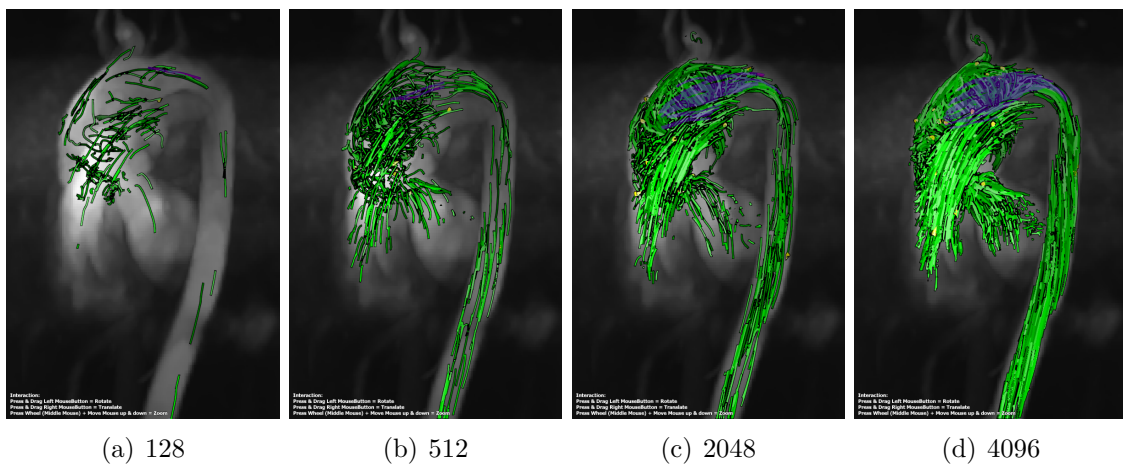


Abbildung 4.7: Ausfüllung des Gefäßes mit Pfadlinien bei unterschiedlicher Partikelzahl

durch einfache binäre Operationen aus den Bereichen mit hohem λ_2 -Wert und der normalen Seeding-Maske berechnet werden. [Abbildung 4.8](#) zeigt eine Darstellung der bei ungleichmäßigen Seeding verwendeten Maske.

Die Anwendung des ungleichmäßigen Seeding stellt sicher, dass ein großer Teil der Partikel in Gebieten erzeugt wird, in denen Wirbel auftreten, während gleichzeitig laminare Bereiche weiterhin visualisiert werden. Profitieren kann hierbei vor allem eine Visualisierung, bei der nur wenig Partikel eingesetzt werden.



Abbildung 4.8: Ungleichmäßiges Seeding: Gleichmäßiger Seeding-Bereich (weiß), Bereiche mit hohem λ_2 -Wert (rot), zusätzliche Seeding-Maske (grün).

4.2.2 Berechnung der Partikelpfade

Die Partikelintegration kann mithilfe des Euler-Verfahrens oder RK4 erfolgen. Verlässt ein Partikel die Gefäßmaske, so wird es entfernt und auf einer zufälligen Position neu erzeugt. Um den Verlauf der Partikel besser darstellen zu können, werden die letzten n Positionen der Partikel gespeichert, um deren Bewegung als Linie darzustellen.

Da die Partikel voneinander unabhängig berechnet werden, wird deren Integration von der Grafikkarte durchgeführt. Im Gegensatz zur CPU können auf der GPU keine Zufallszahlen generiert werden. Dies verhindert die Initialisierung der Partikel an zufälligen Positionen.

Daher werden die Partikel durch die CPU initialisiert und danach auf die GPU hochgeladen. Diese wiederum markiert Partikel, welche die Positionsmaske verlassen haben. Diese werden von der CPU vor dem nächsten Integrationsschritt neu initialisiert. Zu diesem Zweck kopiert diese den für die Partikel verwendeten Speicherbereich von der Grafikkarte in den Arbeitsspeicher und initialisiert entweder alle Partikel (zu Beginn der Visualisierung) oder sucht nach Partikeln, die neu initialisiert werden müssen (nach jedem Integrationsschritt).

Um sicherzustellen, dass die Partikel innerhalb der Anatomie platziert werden, wird eine Liste von geeigneten Voxelpositionen angelegt. Die Entscheidung, ob ein Voxel sich innerhalb der Anatomie befindet, wird mithilfe eines an die Visualisierung gekoppelten Schwellenwerts getroffen. Aus dieser Liste wird nun für jedes Partikel ein zufälliges Element ausgewählt und als Startposition gesetzt. Auch alle vergangenen Positionen des Partikels werden mit diesem Wert initialisiert. Wurden alle Partikel positioniert, wird der Speicherbereich zurück auf die GPU kopiert.

Die Flussberechnung pro Partikel wird innerhalb eines Compute Shaders anhand der Flusstextur durchgeführt. Bevor die neue Position des Partikels gespeichert wird, werden alle vergangenen Positionen in der Liste um eins nach hinten geschoben. Die älteste gespeicherte Position wird verworfen und die neue Position an die erste Stelle gesetzt. Neben der Bewegungsberechnung des Partikels wird in jedem Schritt überprüft, ob dieses die Anatomie verlässt. Ist das der Fall, so wird das Partikel sowie alle vergangenen Positionen auf den Wert $(0, 0, 0)$ gesetzt, was die CPU veranlasst, diesen Partikel neu zu initialisieren.

Der λ_2 -Wert wird bei jedem Integrationsschritt zusätzlich berechnet und zusammen mit der Position des Partikels gespeichert. Hierfür gibt es zwei grundsätzliche Ansätze:

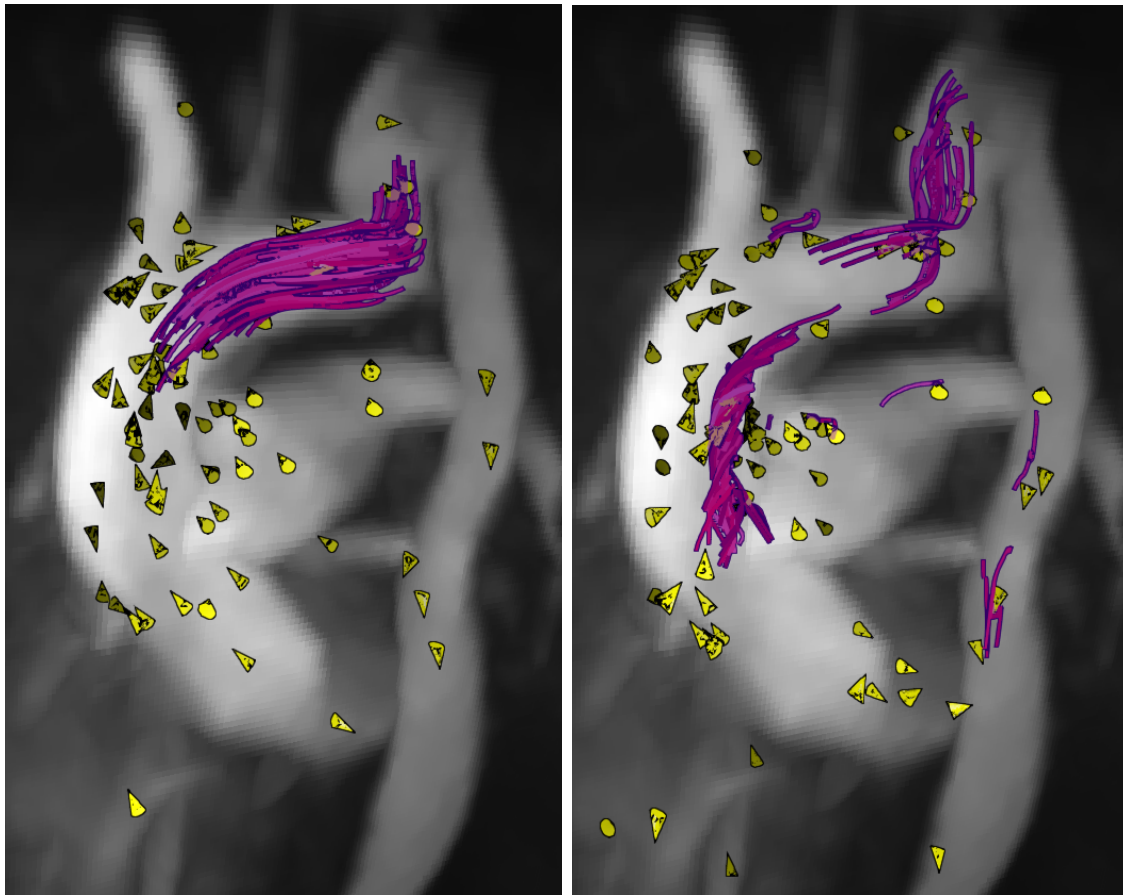
Die erste Möglichkeit ist das in [Abschnitt 4.4.3](#) beschriebene Vorberechnen der λ_2 -Werte für jeden Voxel und jeden Zeitschritt im Datensatz. Hierzu wird ein separater Compute-Shader verwendet. Dieser wird in mehreren Schritten ausgeführt, wobei immer eine Gruppe von Instanzen für alle Voxel pro Zeitschritt eingesetzt wird. Diese Aufteilung ist nötig, damit nicht alle Flusstexturen aller Zeitschritte gleichzeitig geladen werden müssen. Arbeiten alle Instanzen des Shaders jedoch nur an einem einzelnen Zeitschritt zur gleichen Zeit, so müssen bloß die Flusstexturen des aktuellen und jeweils nächsten Zeitschrittes geladen werden. Um letztendlich eine vollständige λ_2 -Textur für alle Zeitschritte zu erhalten, wird eine Schleife verwendet, welche die benötigten Texturen bindet und danach die nächste Gruppe von Shader-Instanzen startet.

Die zweite Möglichkeit ist die Ermittlung der λ_2 -Werte zur Laufzeit zusammen mit der Berechnung der Partikelpositionen. Dieses Verfahren produziert genauere Ergebnisse, da die Berechnung des Wertes an der exakten Position des Partikels durchgeführt und nicht aus zuvor berechneten Werten an den umliegenden Gitterpunkten interpoliert wird. Allerdings verlangsamt die zusätzliche Berechnung der λ_2 -Werte die Partikelvisualisierung und ist sensitiver gegenüber Rauschen.

Der Unterschied zwischen beiden Methoden wird in [Abbildung 4.9](#) deutlich. Die in Echtzeit berechneten λ_2 -Pfadlinien erfassen auch kleinere Wirbel und geben die Position größerer Wirbel genauer wieder. Gleichzeitig erhöhen Sie jedoch die Anfälligkeit der Berechnung gegenüber Rauschen, sodass auch in Bereichen mit laminarem Fluss einzelne Linien als Wirbel eingestuft werden.

4.2.3 Visualisierung der Partikel

Ein naiver Ansatz wäre, die berechneten Positionen aus dem GPU-Speicher zurück zur CPU zu senden, dort anhand der Positionen eine entsprechende Geometrie zu



(a) Wirbelvisualisierung mit vorberechneten λ_2 -Werten (b) Wirbelvisualisierung mit in Echtzeit berechneten λ_2 -Werten

Abbildung 4.9: Vergleich von vorberechneten mit in Echtzeit berechneten λ_2 -Werten. Zu laminarem Fluss gehörende Linien sind ausgeblendet, der Fluss in diesen Bereichen wird stattdessen durch gelbe Richtungspfeile visualisiert.

generieren und diese zur Visualisierung wieder zurück zur GPU zu übertragen. Der wiederholte Datentransfer zwischen CPU und GPU sowie das eigentlich hochgradig parallelisierbare Iterieren des Prozessors über die Partikel macht dieses Verfahren jedoch äußerst ineffizient.

Um den Großteil der Visualisierung von der GPU durchführen zu lassen, bietet es sich an, auf der CPU nur eine minimale Geometrie zu erzeugen, die als Ankerpunkt in den Shadern verwendet wird. Dabei wird für jeden Partikelpfad ein Grafikprimitiv mit einzigartiger ID erzeugt. Bei der Verarbeitung des Grafikprimitivs wird von der Grafikkarte der zu diesem Primitiv gehörende Partikelpfad ermittelt und anstelle des Primitivs visualisiert. Dies findet im Geometry Shader statt, wobei jede Shader-Instanz genau eine Pfadlinie erzeugt. Um die Instanzen zu starten, erzeugt der Prozessor für jede Pfadlinie genau einen Eckpunkt, der als Attribut die ID der entsprechenden Pfadlinie enthält und dem Shader somit erlaubt, die von ihm zu visualisierende Pfadlinie auszuwählen.

Dieser Punkt wird in eine Linie aus mehreren Segmenten umgewandelt. Dazu greift der Geometry Shader auf die bereits auf der GPU in einem Buffer gespeicherten

Informationen über alle Pfadlinien zu und wählt die von ihm zu erzeugende Linie anhand der X-Koordinate in seinem Eingangs-Punkt aus. Zusätzlich werden je nach Benutzereinstellung zu jedem Punkt λ_2 -Flussgeschwindigkeits- oder Rückflussinformationen berechnet und an die Punkte angehängt.

Die erzeugte Geometrie wird nun im Fragment Shader gerendert, wobei die zusätzlich gelieferten Daten als Basis für die Farbe der Liniensegmente verwendet werden. Zudem werden um jeden Pfad herum schwarze Begrenzungslinien (*Silhouetten*) gezeichnet, um die Wahrnehmung von Verdeckungsverhältnissen zu verbessern (siehe [Abschnitt 4.1.2.3](#)). Da diese auf Tiefenwerten basiert, verschwinden die Begrenzungslinien, wenn die Pfade zu dicht beieinander liegen. Um die unterschiedlichen Linien dennoch optisch unterscheiden zu können, wird die Helligkeit jeder einzelnen Linie angepasst. Art und Stärke der Anpassung wird anhand der letzten Ziffer der ID der Linie ermittelt. Endet die ID auf 0, so wird die Helligkeit der Linie um 50 % verringert, bei 9 um 50% erhöht. Zwischen diesen Werten wird entsprechend interpoliert. Der Einfluss der unterschiedlichen Techniken wird in [Abbildung 4.10](#) dargestellt.

Um die Sichtbarkeit innenliegender Flusstrukturen zu erhöhen, kann ein Overlay zugeschaltet werden. Dieses legt sich sowohl über die Gefäß- als auch über die Partikelsimulation und weist den Nutzer darauf hin, an welchen Stellen sich verdeckte Flusstrukturen befinden. Die Berechnung dieses Overlays findet im Fragment Shader statt. Befindet sich an einer Position ein Wirbel, so wird der entsprechende Pixel in einer Textur notiert. Erst danach wird der Tiefentest durchgeführt, der zum Verwerfen des entsprechenden Fragments führen könnte. Nachdem das Rendering von Partikeln und Gefäßvisualisierung abgeschlossen ist, wird die Overlay-Textur semi-

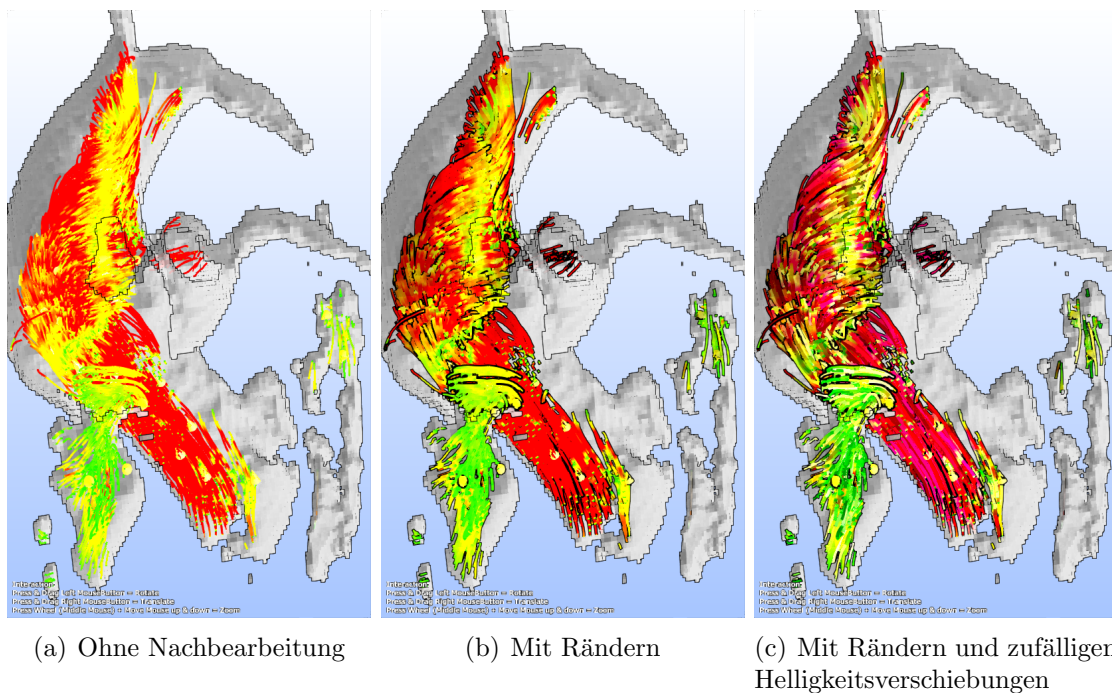


Abbildung 4.10: Vergleich der verschiedenen Nachbearbeitungsschritte beim Zeichnen der Pfadlinien.

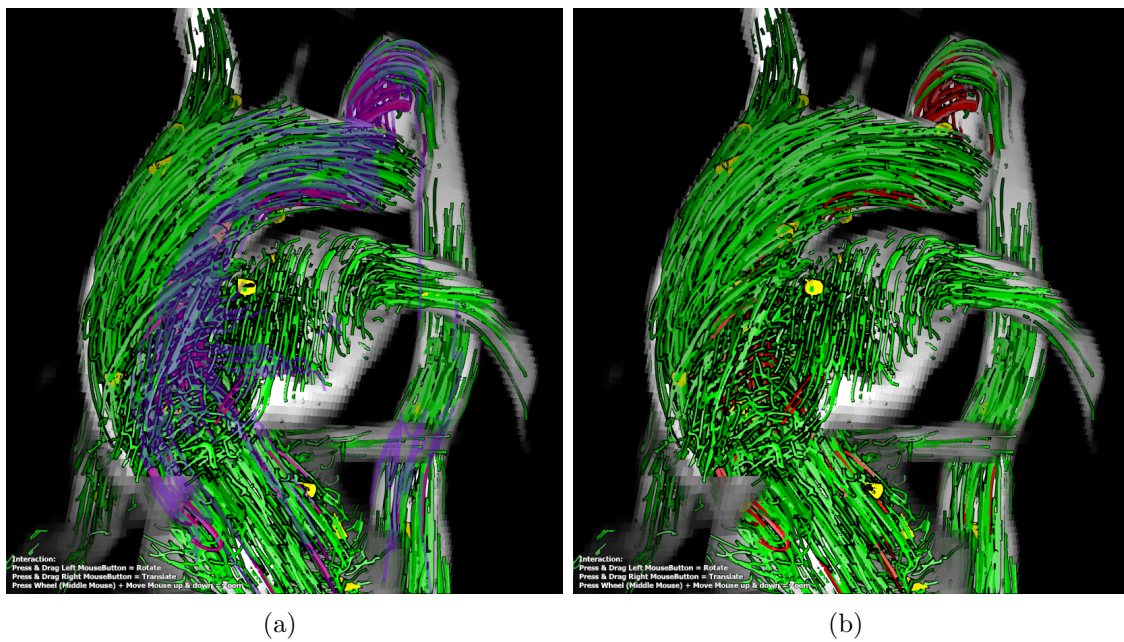


Abbildung 4.11: Partikelvisualisierung (a) mit und (b) ohne Overlay. Innenliegende Teile des Wirbels sind ohne Overlay schlechter erkennbar.

transparent über das gerenderte Bild gezeichnet. [Abbildung 4.11](#) zeigt den Vergleich zwischen Partikelvisualisierung mit und ohne Berechnung des Overlays.

Alternativ zur Aktivierung der Overlays können auch Pfadlinien in laminarem Fluss ganz oder teilweise entfernt werden. Dies gibt den Blick auf innenliegende Wirbelstrukturen frei. Das Ausmaß der Entfernung kann vom Anwender eingestellt werden. [Abbildung 4.12](#) zeigt einen Vergleich zwischen verschiedenen Entfernungsstärken. Ähnlich wie bei der Erzeugung des Overlays wird auch hier im Fragment Shader getestet, ob ein Linienfragment einen Wirbel enthält. Ist dies nicht der Fall, so wird anhand der ID der Linie sowie der Einstellungen des Nutzers überprüft, ob die Linie durch ein Verwerfen aller ihrer Fragmente entfernt werden soll.

Zusätzlich zu den Pfadlinien werden Richtungsindikatoren in Form von gelben Pfeilen erzeugt. Diese bewegen sich zusammen mit den Pfadlinien durch das Gefäß und

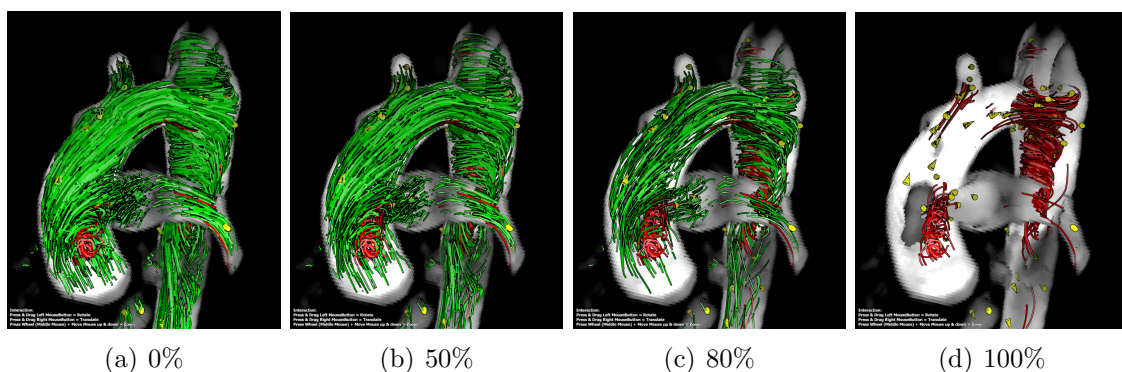


Abbildung 4.12: Entfernung einer unterschiedlichen Anzahl von Pfadlinien in laminarem Fluss.

zeigen dabei jederzeit in Richtung des Flusses. Die Positionierung dieser Pfeile basiert auf Partikelpositionen. An der Position jedes 10. Partikels wird ein solcher Pfeil gezeichnet. In [Abbildung 4.9](#) und [Abbildung 4.12](#) sind diese Pfeile gut zu erkennen.

4.3 Komposition von Partikeln und DVR

Mit der über DVR visualisierten Anatomie und den aus Grafikprimitiven gerenderten Partikelpfaden liegen zwei grundverschiedene Visualisierungen vor, die für die Ausgabe zusammengeführt werden müssen. Dazu könnten die Partikel einfach über der Anatomie gezeichnet werden. Jedoch verhindern fehlende Tiefeninformationen, dass beide Darstellungen als Einheit wahrgenommen werden. Es ist demzufolge nötig, beide Visualisierungen miteinander zu integrieren.

Dies stellt jedoch ein Problem dar, da die Grafikhardware keine Standardfunktionen anbietet, um Grafikprimitive in ein mittels DVR gerendertes Objekt zu integrieren. Eine potentielle Lösung dieses Problems wäre die Voxelisierung der Partikel. Dies würde jedoch erhebliche Leistungseinbußen bedeuten sowie eine große Menge zusätzlichen Speichers konsumieren, da die Voxelisierung bei jedem Rendervorgang von neuem durchgeführt werden müsste.

4.3.1 Komposition anhand der Tiefenwerte

Mithilfe des Z-Buffers, der wie in [Abschnitt 4.1.4](#) beschrieben während des DVR generiert wurde, ist eine Komposition beider Visualisierungen ohne großen Rechenaufwand möglich. Die Tiefenwerte aus dem Z-Buffer werden lediglich mit der Partikelvisualisierung verglichen, um festzustellen, welche der Visualisierungen auf jedem Pixel angezeigt werden soll. Für die MIP-Darstellung ist dieses Vorgehen jedoch nicht möglich, da beim Rendern der Anatomie sämtliche Tiefeninformationen verworfen werden.

Bei der erweiterten Rampenfunktion ist eine differenziertere Betrachtung erforderlich. Dort kann es aufgrund der Verwendung von Transparenz vorkommen, dass ein Partikel nur teilweise von der Anatomie verdeckt wird. Aus diesem Grund wird die Komposition schon während des Renderns der Anatomie durchgeführt. Bei jedem Schritt auf dem Strahl wird der aktuelle Tiefenwert des Punktes mit demjenigen aus dem Z-Buffer der Partikelvisualisierung verglichen. Schneidet der Strahl einen Partikelpfad, so wird das Rendering an dieser Stelle abgebrochen und die bisher akkumulierten Farb- und Opazitätswerte mit der Farbe des Partikelpfades verrechnet, um einen Überdeckungseffekt zu produzieren.

Bei der soliden Darstellung ist eine Komposition weniger komplex, da weder die Partikelpfade noch die Oberfläche transparent dargestellt werden. Kreuzt ein Partikelpfad den Strahl vor der ersten Überschreitung des Schwellenwertes der Visualisierung, so wird lediglich die Farbe des Partikels auf diesem Pixel gezeichnet. Andernfalls wird die Partikelfarbe mit einer Transparenz von 90% dargestellt. Dies stellt sicher, dass die Anatomie Partikelpfade nicht vollständig verdeckt.

Ein Nachteil dieses Vorgehens ist, dass semitransparente Partikelpfade nicht dargestellt werden können. Der Z-Buffer enthält ausschließlich Tiefeninformationen über

das dem Betrachter am nächsten liegende Objekt - dahinterliegende Strukturen verfügen nicht über Tiefenwerte und können demzufolge auch nicht mit der DVR-Visualisierung kombiniert werden.

Dieses Problem könnte durch den Einsatz von *Order Independent Transparency* (OIT) gelöst werden. Ursprünglich wurde dieses Verfahren entwickelt, um halbtransparente Grafikprimitive anzuzeigen, die sich gegenseitig überdecken, ohne diese zuvor nach ihrer Entfernung zum Betrachter sortieren zu müssen. Diese Sortierung ist erforderlich, da die Überdeckungen nur dann korrekt berechnet werden, wenn weiter entfernte Objekte zuerst gezeichnet werden. Bei komplexen Szenen ist diese Sortierung jedoch aufwendig und auch nicht immer eindeutig möglich [Kub14].

Die Grundidee von OIT ist es, während des Render-Vorgangs im Fragment Shader eine sortierte Liste aller überdeckenden Flächen pro Pixel zusammen mit ihren Transparenz- und Farbwerten zu erzeugen und diese anschließend zu akkumulieren [YHGT10]. Interpretiert man die Sample-Punkte beim DVR als einzelne, semitransparente Flächen, so könnte OIT eingesetzt werden, um semitransparente Partikelpfade zu ermöglichen. Da allerdings das dynamische Allokieren von Speicher innerhalb der GPU-Shader nicht möglich ist, muss die Größe der OIT-Liste im Vorfeld festgelegt werden und kann sich nicht an der tatsächlichen Anzahl überlappender Flächen pro Pixel orientieren. Somit muss für jeden Pixel genug Speicher reserviert werden, um die maximal mögliche Zahl an Sampling-Punkten sowie Überlappungen der Partikelpfade zu speichern. Geht man pro Listeneintrag von 96 Bit (Farbe, Transparenz und Tiefe mit je 32 Bit) sowie einer Auflösung von 800 x 600 Pixeln aus, müssten etwa 5.5 MB für jede Ebene der Liste gespeichert werden. Bei hunderten von Sample-Punkten und tausenden von einzelnen Partikelpfaden würde hier eine unverhältnismäßige Belastung des Grafikspeichers entstehen, was den Einsatz von OIT verhindert.

4.3.2 DVR-Frontface-Culling

Durch die Komposition anhand der Tiefenwerte ist die von der Partikelvisualisierung generierte Geometrie zwar nun in das DVR integriert, allerdings werden die Partikel größtenteils von der Anatomie verdeckt, da sie sich in ihr befinden. Dies ist im zweiten Bild in [Abbildung 4.13](#) dargestellt. Würde die Anatomie über ein Dreiecksnetz gerendert werden, so könnte dieses mithilfe des Frontface-Cullings geöffnet werden, sodass nur die Rückseiten der Anatomie gezeichnet werden und die Partikel somit freiliegen.

Da beim DVR keine expliziten Oberflächen vorhanden sind, ist es erforderlich, die Rückseite der Anatomie zu finden. Dazu wird derjenige Punkt auf dem Strahl gesucht, an dem die Intensität zum ersten Mal einen bestimmten Schwellenwert unterschreitet, nachdem dieser zuvor überschritten wurde. Beim soliden Rendering ist dieser Schwellenwert bereits durch die Visualisierung gegeben, beim Verwenden der erweiterten Rampenfunktion bietet sich der Startwert der Rampe (b_1) an. Alle Sample-Punkte, die vor diesem Punkt ausgelesen wurden, werden nun beim Rendering ignoriert, sodass nur die Rückwand des Gefäßes dargestellt wird.

Dieses Vorgehen hat jedoch den Nachteil, dass die Rückwand nun mit nur einem einzelnen Voxel dargestellt wird, was das Verfahren anfällig gegenüber Rauschen

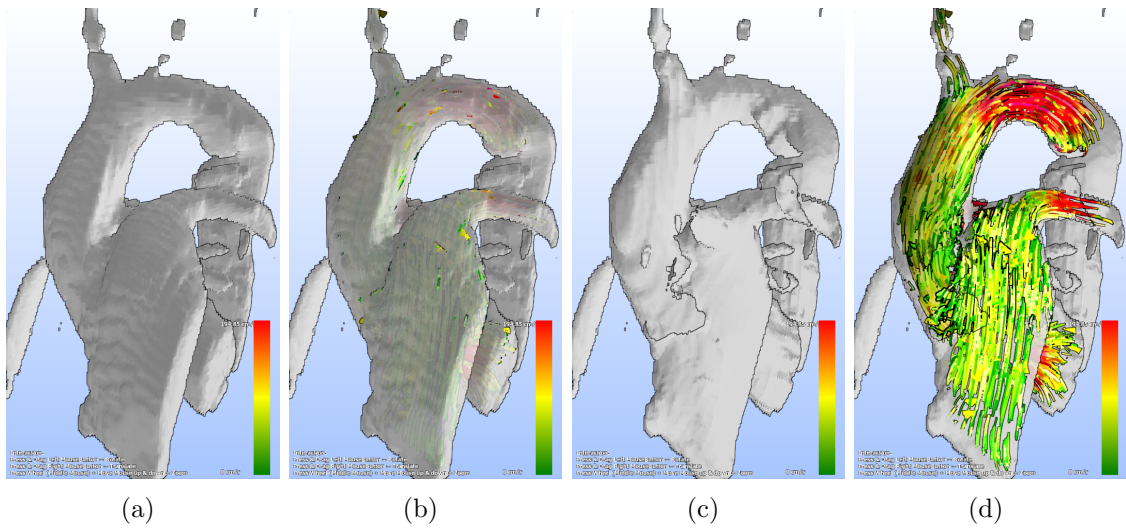


Abbildung 4.13: (a) Visualisierung des Gefäßes ohne Partikel und Frontface-Culling; (b) durch Gefäßanatomie verdeckte Partikel; (c) Rückwand des Gefäßes; (d) durch Frontface-Culling sichtbare Partikel.

macht. Aus diesem Grund wird nicht auf einen Schwellenwertübergang zwischen zwei Sample-Punkten, sondern innerhalb eines ganzen Bereichs untersucht. Dieser umfasst alle zukünftigen Sample-Punkte in einer Entfernung von maximal 4.5 mm, mindestens jedoch 5 Punkte. Bei minimaler Distanz der Sampling-Werte werden 15 Sampling-Punkte verwendet. Unterschreiten mindestens 55% der gemessenen Werte den Schwellenwert, so wird der aktuelle Sample-Punkt als Teil einer Rückwand markiert und gezeichnet, andernfalls wird er übersprungen. Durch die Länge des untersuchten Abschnittes (4.5 mm) sowie den Schwellenwert von 55% ergibt sich, dass die gezeichneten Rückwände eine ungefähre Dicke von 2 mm haben, was der durchschnittlichen Dicke der Aortenwand entspricht [LKR+04]. Die Mindestzahl von 5 Sampling-Punkten garantiert, dass die Ermittlung der Rückwand auch bei größerer Sampling-Distanz relativ unempfindlich gegenüber Rauschen bleibt. [Abbildung 4.14](#) zeigt das grundlegende Funktionsprinzip dieses Ansatzes.

[Abbildung 4.13](#) zeigt den Effekt des Frontface-Cullings mit (4. Bild) und ohne (3. Bild) Partikel.

Das Ausblenden nicht zur Rückwand gehörender Voxel erfolgt je nach gewählter Visualisierung unterschiedlich. Bei der erweiterten Rampenfunktion wird dies durch eine Manipulation der Akkumulationsformel erreicht:

$$O_p = O_{p-1} \cdot (1 - (o_p \cdot c)) \quad G_p = G_{p-1} + O_p \cdot g_p \quad (4.4)$$

Der neu eingeführte Parameter c fungiert als Korrekturfaktor für das Frontface-Culling. Ist dieses aktiviert, so wird geprüft, ob der aktuell untersuchte Sample-Punkt in der Rückwand eines Gefäßes liegt. Ist dies nicht der Fall, nimmt c den Wert 0 an, sodass der aktuelle Punkt unsichtbar wird. Andernfalls hat c den Wert 1 und somit keinen Einfluss auf die Berechnung.

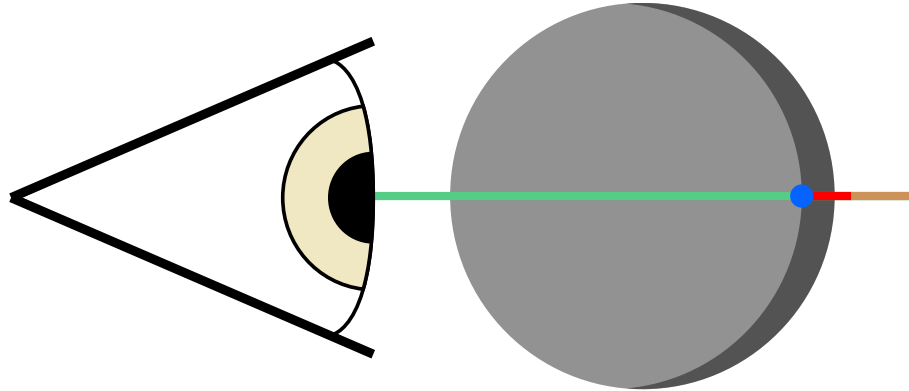


Abbildung 4.14: Frontface-Culling; Aktueller Punkt (blau) mit Analysebereich (rot). Alle Voxel vor dem aktuellen Punkt werden mit reduzierter Opazität gezeichnet (grüner Strahl). Da sich die letzten 10% des Analysebereichs unterhalb des Schwellenwertes (grauer Kreis) befinden, werden alle nachfolgenden Voxel mit voller Opazität gezeichnet.

4.4 Vorberechnungen

Während des Ladens der Datensätze werden einige Vorberechnungen durchgeführt, welche die Qualität der eingelesenen Daten verbessern oder die Echtzeitvisualisierung entlasten.

4.4.1 Verbesserung der Flussbilder

4D PC-MRI Daten enthalten starkes Rauschen in luftgefüllten Räumen wie der Lunge. Obwohl diese sich nicht mit den Gefäßen überschneiden, können sie einen negativen Einfluss auf die Visualisierungsqualität haben. Histogrammbasierte Funktionen wie die Ermittlung der maximalen Flussgeschwindigkeit werden durch diese Ausreißer gestört.

Derartige Artefakte können aus den Daten mittels der Berechnung der temporalen Standardabweichung ($tStDev$) erkannt und herausgefiltert werden. Während sich die Flussvektoren in den Gefäßen über die Zeit eher langsam ändern, weisen die Artefaktdaten große Unterschiede selbst zwischen direkt aufeinanderfolgenden Zeitpunkten auf. Die Standardabweichung der mit Artefakten belasteten Bereiche ist somit deutlich höher als jene innerhalb des Gefäßes. Über die Anwendung eines Schwellenwertes können nun die gestörten Bereiche identifiziert und herausgefiltert werden.

4.4.2 Flussrichtung

Um in der Partikelvisualisierung Rückflüsse markieren zu können, wird für jeden Voxel im Datensatz eine Hauptflussrichtung als Durchschnitt aller Flussvektoren gebildet. Schnellerer Fluss, der primär während der Systole vorhanden ist, wird dabei implizit höher gewichtet. Dabei werden die 4D-Flussbilder auf drei Dimensionen reduziert. Die entstandene 3D-Textur kann von der Partikelvisualisierung herangezogen werden, um Rückflüsse über einen negativen Winkel zur Hauptflussrichtung zu erkennen.

Problematisch sind Bereiche mit äußerst geringen Flussgeschwindigkeiten aufgrund des schlechten Signal-Rausch-Verhältnisses. Auch Verwirbelungen stellen ein Problem dar, da deren Hauptflussrichtung oft nicht parallel zur Centerline des Gefäßes verläuft.

4.4.3 Wirbelkriterium

Um die Grafikhardware zu entlasten, kann die Berechnung des λ_2 -Wirbelkriteriums bereits beim Laden des Datensatzes erfolgen. Für jeden Zeitschritt wird eine 3D λ_2 -Textur durch die Anwendung des Wirbelkriteriums auf den Mittelpunkt jedes Voxels erzeugt. Bei der Partikelvisualisierung ist nun eine λ_2 -Berechnung pro Partikel nicht mehr erforderlich, da der λ_2 -Wert aus der Textur interpoliert wird.

Dies beschleunigt zwar die Simulation, verringert jedoch auch deren Qualität, da es sich hierbei um eine *preclassification*-Methode handelt. Alternativ steht dem Anwender auch eine *postclassification*-Variante der λ_2 -Berechnung zur Verfügung. Hier wird das λ_2 -Kriterium während der Visualisierung in Echtzeit aus interpolierten Flussdaten berechnet.

5. Ergebnisse und Diskussion

Im folgenden Kapitel werden sowohl die für die Evaluation verwendeten Datensätze als auch deren Ergebnisse behandelt. Der Fokus hierbei liegt auf der Probandenstudie, die sowohl Stärken als auch Schwächen der in dieser Arbeit verwendeten Methode aufzeigt. Auch die Performance des Prototypen wird in diesem Kapitel behandelt.

5.1 Datensätze

Für die Entwicklung und Evaluierung des Prototypen stehen 16 Datensätze zur Verfügung, die sowohl gesunde Probanden als auch Patienten mit verschiedenen Pathologien umfassen. Eine Übersicht über alle Datensätze befindet sich in [Tabelle 5.1](#).

Datensatz	Pathologie	Auflösung
2_2012_Fallot	Fallot-Tetralogie	$132 \times 192 \times 23 \times 14$ (1.77 mm $\times$ 1.77 mm $\times$ 3.5 mm $\times$ 49.2 ms)
anonymous	Keine	$132 \times 192 \times 24 \times 17$ (1.77 mm $\times$ 1.77 mm $\times$ 3.5 mm $\times$ 49.2 ms)
Aorta_13_2012	Keine	$132 \times 192 \times 22 \times 16$ (1.77 mm $\times$ 1.77 mm $\times$ 3.5 mm $\times$ 49.2 ms)
Aortenbypass_1_2011	Stenose und Bypass	$132 \times 192 \times 15 \times 15$ (1.77 mm $\times$ 1.77 mm $\times$ 3.5 mm $\times$ 49.2 ms)
Aortenektasie_04_2011	Ektasie	$132 \times 192 \times 25 \times 18$ (1.77 mm $\times$ 1.77 mm $\times$ 3.5 mm $\times$ 49.1 ms)
Aortenektasie_09_2012	Ektasie	$132 \times 192 \times 18 \times 16$ (1.77 mm $\times$ 1.77 mm $\times$ 4 mm $\times$ 55.7 ms)
Flow_MD_1_2014	Keine	$132 \times 192 \times 20 \times 15$ (1.77 mm $\times$ 1.77 mm $\times$ 3 mm $\times$ 49.1 ms)
Flow_MD_2_2014	Ektasie	$132 \times 192 \times 20 \times 24$ (1.77 mm $\times$ 1.77 mm $\times$ 3 mm $\times$ 49.2 ms)
Flow_MD_3_2014	Aortenklappenstenose	$132 \times 192 \times 20 \times 11$ (1.77 mm $\times$ 1.77 mm $\times$ 3 mm $\times$ 48.8 ms)
Flow_MD_4_2014	Aortenklappenstenose, Ektasie	$132 \times 192 \times 20 \times 13$ (1.77 mm $\times$ 1.77 mm $\times$ 3 mm $\times$ 47.7 ms)
Flow_MD_7_2014	Fallot-Tetralogie, Pulmonalstenose	$132 \times 192 \times 24 \times 17$ (1.77 mm $\times$ 1.77 mm $\times$ 3 mm $\times$ 49.2 ms)
ISTA_06_2012	ISTA	$132 \times 192 \times 22 \times 21$ (1.77 mm $\times$ 1.77 mm $\times$ 3.5 mm $\times$ 49.2 ms)
ISTA_08_2012	ISTA	$132 \times 192 \times 18 \times 15$ (1.77 mm $\times$ 1.77 mm $\times$ 3.5 mm $\times$ 57.8 ms)
ISTA_12_2012	Aneurysma	$132 \times 192 \times 15 \times 12$ (1.77 mm $\times$ 1.77 mm $\times$ 3.5 mm $\times$ 49.2 ms)
ProtheseISTA_11_2012	ISTA und Prothese	$132 \times 192 \times 15 \times 19$ (1.77 mm $\times$ 1.77 mm $\times$ 3.5 mm $\times$ 57.8 ms)
vierDFlussUp	Keine	$132 \times 192 \times 20 \times 18$ (1.77 mm $\times$ 1.77 mm $\times$ 3.5 mm $\times$ 49.2 ms)

Tabelle 5.1: Übersicht aller verwendeten Datensätze.

Diese Daten stammen vom Herzzentrum der Universität Leipzig sowie dem Klinikum Magdeburg. Die Akquirierung erfolgte mit einem 3 Tesla Magnetom Verio MR (Siemens Healthcare, Erlangen), der V_{ENC} war in allen Fällen auf 1.5 m/s festgelegt.

5.2 Visualisierung der Anatomie

Die Gefäßstruktur ließ sich in den meisten Datensätzen mit keinen oder nur wenigen Störungen visualisieren. [Abbildung 5.1](#) zeigt den Datensatz *Aortenbypass_1_2011* als Beispiel für eine gut funktionierende Visualisierung. Da das Blut während des Herzzyklus in allen Bereichen der Gefäße mit ausreichender Geschwindigkeit fließt, weisen diese einen sehr hohen Kontrast im Vergleich zum Hintergrund auf. Somit sind sowohl die Aorta als auch die Pulmonalarterie vollständig abgebildet und gut erkennbar. Ebenfalls dargestellt wird ein Teil der Ventrikel. Dies ist nicht explizit gewollt, lässt sich aber nicht vermeiden, da in Teilen des Herzens ähnliche Flussgeschwindigkeiten auftreten wie in Bereichen der Aorta und Pulmonalarterie. Dies lässt sich gut in der MIP-Darstellung des Datensatzes erkennen. Bei der histogrammbasierten Visualisierung durch Rampenfunktion und solide Körper sind diese Teile demzufolge ebenfalls sichtbar.

Problematisch wird die Darstellung des Ventrikels jedoch, wenn sehr viel davon zu sehen ist und andere Bereiche somit verdeckt werden. Ein Beispiel dafür ist der Datensatz *2_2012_Fallot* in [Abbildung 5.2](#), bei dem ein sehr großer Teil des rechten Ventrikels sichtbar ist. Dieser macht es schwierig, die Pulmonalarterie korrekt zu erkennen. Wird die Visualisierung jedoch so eingestellt, dass der Ventrikel nicht mehr sichtbar ist, so verschwindet auch ein großer Teil von Aorta und Pulmonalarterie, da diese ähnliche Intensitätswerte aufweisen.

Ein ebenfalls häufig auftretendes Artefakt ist das Ausblenden eines Teils des Aortenbogens. Dies ist in [Abbildung 5.3](#) dargestellt. In der MIP-Darstellung ist erkennbar, dass dieser Teil einen reduzierten Kontrast im Vergleich zum Rest des Gefäßes aufweist. Dies deutet auf eine reduzierte Flussgeschwindigkeit in diesem Bereich hin, die daraus resultieren könnte, dass der Arm-Kopf-Gefäßstamm, die linke Halsschlagader sowie die linke Unterschlüsselbeinarterie an dieser Stelle von der Aorta abgehen.

Ein Beispiel für eine nicht funktionierende Visualisierung stellt *Flow_MD_3_2014* in [Abbildung 5.4](#) dar. Der Datensatz ist mit Artefakten behaftet, die eine Visualisierung ohne vorhergehende Segmentierung stark erschweren. Im linken Teil des Datensatzes befinden sich plattenartige Artefakte mit hoher Intensität, die eine automatische Festlegung von Schwellenwerten bzw. Transferfunktionen aufgrund einer Verfälschung des Histogramms stark erschweren. Aus diesem Grund sind die Gefäße auf den automatischen Visualisierungen kaum zu erkennen. Gleichzeitig weisen die Gefäßstrukturen einen sehr geringen Kontrast auf, was besonders im Aortenbogen deutlich wird. Bei einer manuellen Anpassung von Schwellenwerten und Transferfunktionen werden somit auch verstärkt Bereiche gerendert, die nicht zum Gefäß gehören. Eine annehmbare Visualisierung des Datensatzes mittels Rampenfunktion oder solider Körper ist somit weder automatisch noch mit manueller Einstellung möglich. In der MIP-Darstellung hingegen bleibt die Aorta erkennbar.

Ein ähnliches Verhalten zeigt der Datensatz *ProtheseISTA_11_2012*, der in [Abbildung 5.5](#) dargestellt wird. Die geringere Flussgeschwindigkeit im Aortenbogen führt zu einer Reduzierung des Kontrastes. In der absteigenden Aorta fließt das Blut wieder mit höherer Geschwindigkeit, sodass diese einen besseren Kontrast aufweist. Auch hier kann eine manuelle Anpassung der Transferfunktion bzw. Schwellenwerte

die Darstellung nicht verbessern, da sehr viele nicht zum Gefäß gehörende Bereiche ähnliche Intensitätswerte aufweisen wie der Großteil des Aortenbogens.

Der Datensatz *Flow_MD_4_2014* in [Abbildung 5.6](#) weist ein ähnliches Problem auf. Hier zeigt sich ein stark ausgeprägter Inflow-Jet (besonders gut durch seine hohe Intensität in der MIP-Darstellung erkennbar), der vornehmlich an der Aortenwand entlang fließt, während die Geschwindigkeit im Rest des Gefäßes reduziert ist. In der absteigenden Aorta füllt der schnelle Fluss wieder das komplette Gefäß aus, sodass diese besser dargestellt werden kann.

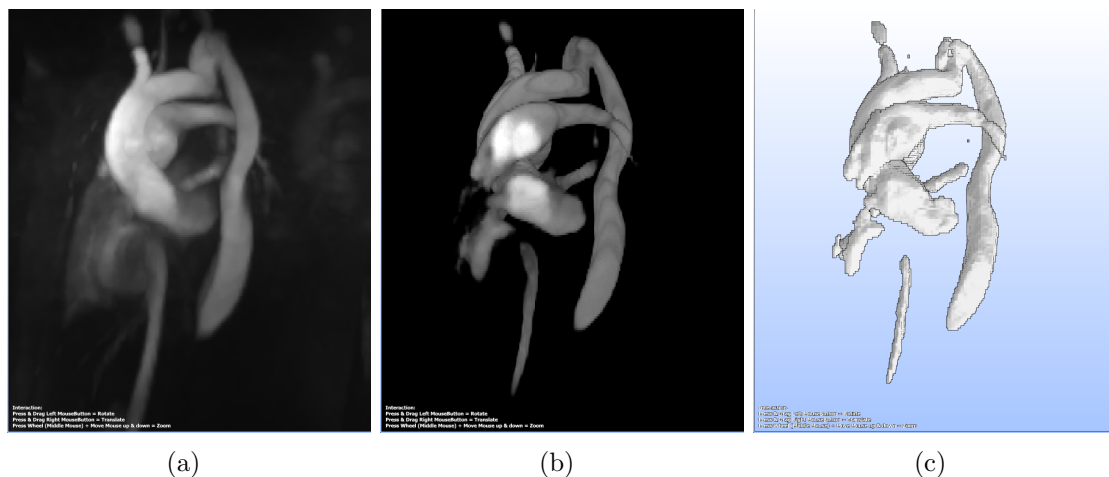


Abbildung 5.1: Visualisierung des Datensatzes *Aortenbypass_1_2011* in den drei Varianten MIP (a), Rampenfunktion (b) und solider Körper (c)

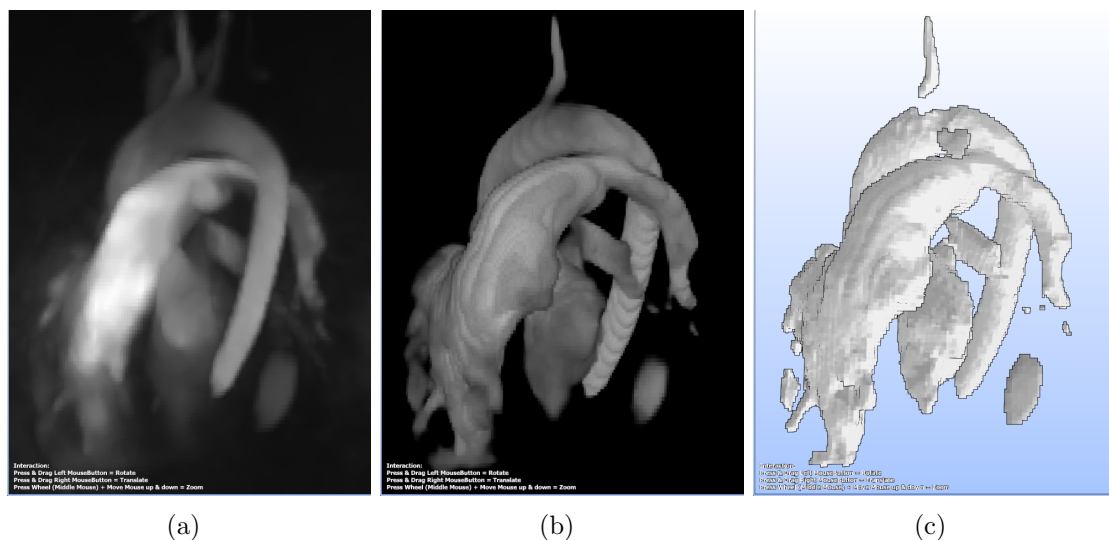


Abbildung 5.2: Visualisierung des Datensatzes *2_2012_Fallot* in den drei Varianten MIP (a), Rampenfunktion (b) und solider Körper (c). Deutlich erkennbar sind Teile des rechten Ventrikels unten links im Bild.

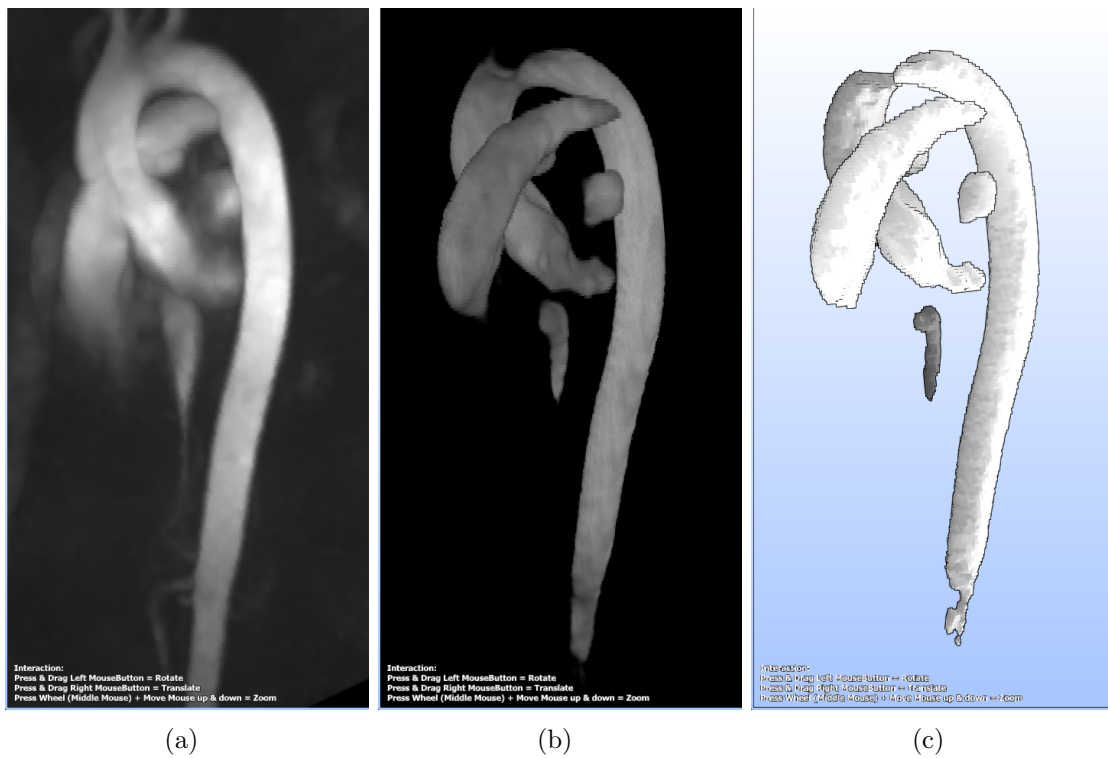


Abbildung 5.3: Visualisierung des Datensatzes *Flow_MD_1_2014* in den drei Varianten MIP (a), Rampenfunktion (b) und solider Körper (c).

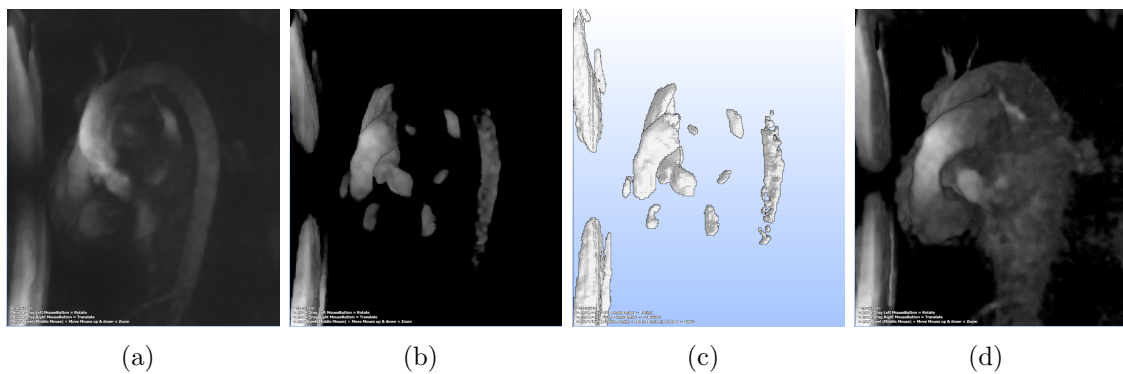


Abbildung 5.4: Visualisierung des Datensatzes *Flow_MD_3_2014* in den drei Varianten MIP (a), Rampenfunktion (b) und solider Körper (c). Visualisierung des Datensatzes mit manueller Transferfunktion (d). Deutlich erkennbar sind die Artefakte im linken Teil der Bilder sowie der fehlende Aortenbogen in (b) und (c).

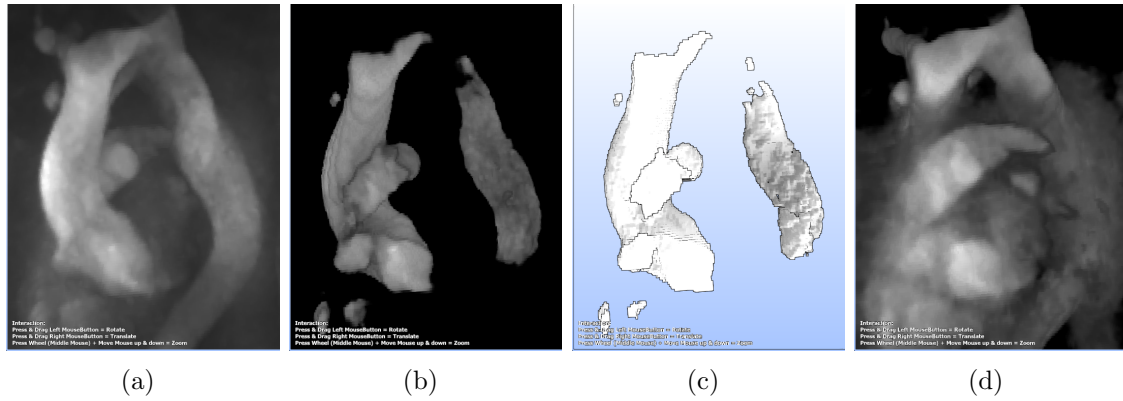


Abbildung 5.5: Visualisierung des Datensatzes *ProtheseISTA_11_2012* in den drei Varianten MIP (a), Rampenfunktion (b) und solider Körper (c). Visualisierung des Datensatzes mit manueller Transferfunktion (d).

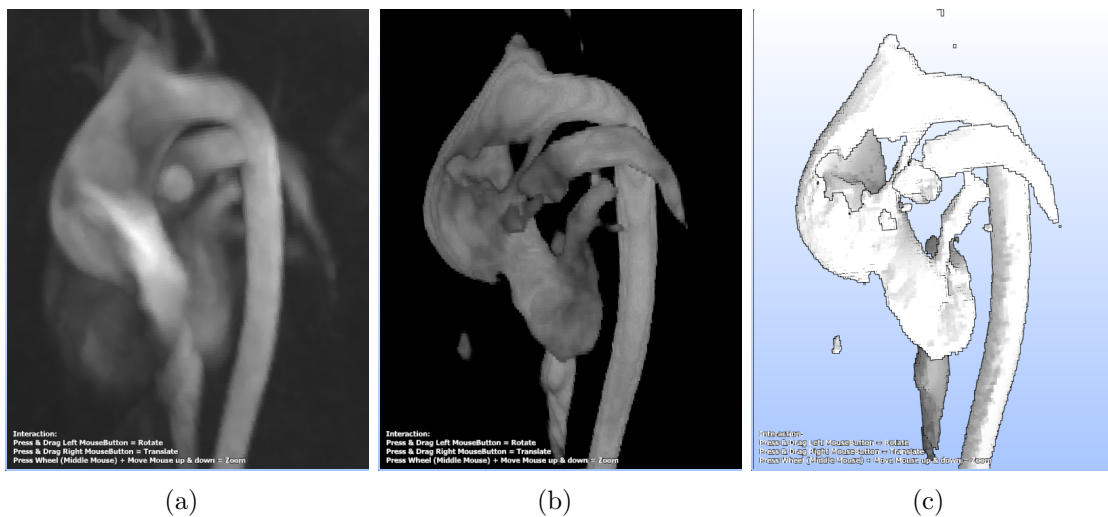


Abbildung 5.6: Visualisierung des Datensatzes *Flow_MD_4_2014* in den drei Varianten MIP (a), Rampenfunktion (b) und solider Körper (c).

5.3 Untersuchung der Datensätze

Um die Visualisierung des Blutflusses sowie die Hervorhebung von Verwirbelungen durch den Prototypen zu evaluieren, wurde ein Test durchgeführt, bei dem Freiwillige einen Teil der zur Verfügung stehenden Datensätze mithilfe des Prototypen untersuchten. Ihre Aufgabe war es dabei, sämtliche vorhandenen Verwirbelungen zu finden und auf einer Anatomiekarte zu skizzieren.

5.3.1 Versuchsaufbau

An der Evaluierung des Prototypen haben acht Freiwillige teilgenommen. Dabei handelte es sich um Studenten und Mitarbeiter der Fakultät für Informatik an der Otto-von-Guericke-Universität Magdeburg. Nach einer kurzen Einführung in die Thematik wurde den Teilnehmern eine spezielle, für den Test angepasste Version des Prototypen mit 13 aufeinander folgenden Datensätzen gezeigt. Zu jedem Datensatz mussten die in der Partikelvisualisierung dargestellten Wirbel in eine Skizze eingezeichnet werden. Einige dieser Skizzen sind in [Abbildung 5.10](#) dargestellt. Die darauf abgebildete Aorta sowie Pulmonalarterie stammen aus demselben Datensatz und wurden manuell segmentiert. Zudem wurden die Teilnehmer gebeten, die Visualisierung des Prototypen mit der Skizze zu vergleichen und zu bewerten, inwieweit die automatische Darstellung mit der manuellen Segmentierung übereinstimmt. Außerdem sollten sie angeben, ob sie durch reines Betrachten der Flussverläufe Verwirbelungen im Datensatz gefunden haben, welche durch die Wirbelextraktion nicht hervorgehoben wurden.

Für jeden Datensatz wurde eine andere Konfiguration des Prototyps verwendet. Diese Konfigurationen unterschieden sich in:

- ihrer Darstellungsart (erweiterte Rampenfunktion, solider Körper und MIP)
- der Art der λ_2 -Berechnung (Vorberechnung und Echtzeit)
- der Verwendung von Overlays für innenliegende Wirbel (an oder aus)

Die Teilnehmer wurden in 4 Gruppen zu je 2 Personen eingeteilt. Allen Gruppen wurden dieselben 13 Datensätze gezeigt, jedoch in unterschiedlicher Reihenfolge und Konfiguration. Lediglich die Darstellungsart war für die Datensätze zwischen den Gruppen konsistent. Durch die Randomisierung der Reihenfolge sollte der Einfluss von Lerneffekten verringert werden.

Der erste Datensatz diente als Übung, um die Teilnehmer mit der Visualisierung und Interaktion vertraut zu machen. Dieser war bei allen Gruppen identisch. Aus den 12 verbliebenen Datensätzen wurden 4 gleichgroße Gruppen gebildet. Bei der ersten Gruppe handelt es sich um Probandendatensätze, die anderen Gruppen erhalten ausschließlich Patientendatensätze. Die Patientendatensätze wurden je nach Bildqualität in einfache, mittlere und schwere Datensätze eingeteilt. Einfache Datensätze zeichnen sich dabei durch eine größtenteils vollständige Darstellung der Anatomie sowie leicht erkennbare Verwirbelungen aus. Bei mittleren Datensätzen treten leichtere Artefakte in der Anatomievisualisierung auf und manche Verwirbelungen erfordern ein manuelles Anpassen des λ_2 -Schwellenwerts. Schwere Datensätze

weisen stärkere Fehler in der Anatomiedarstellung, wie beispielsweise fehlende Aortenbögen, auf.

Die Zuteilung der Datensätze zu Darstellungsarten erfolgte so, dass jede Darstellungsart mit je einem Datensatz aus allen vier Gruppen verwendet wurde. Auf diese Art sollte der Einfluss der Datensatzauswahl auf die Ergebnisse reduziert werden.

Ein Zeitlimit für die Bearbeitung wurde den Teilnehmern nicht gestellt, allerdings wurde die Bearbeitungszeit pro Datensatz erfasst. Die Möglichkeiten zur Anpassung der Visualisierung waren in der für den Test verwendeten Version des Prototypen stark limitiert. Es war lediglich möglich, Schwellenwert bzw. Transferfunktion der Darstellungsart sowie den Schwellenwert für die λ_2 -Wirbelextraktion zu modifizieren. Außerdem konnten sie die Geschwindigkeit der Partikelvisualisierung steuern sowie durch Rotation, Translation und Zoom frei im Datensatz navigieren. Im Anschluss an die Bearbeitung der Datensätze wurden den Teilnehmern allgemeine Fragen zur Bewertung des Prototypen gestellt.

5.3.2 Dauer der Untersuchungen

Insgesamt benötigten die Teilnehmer für die Auswertung jedes Datensatzes durchschnittlich 2:20 min. Der Datensatz *Flow_MD_1_2014* ist mit 1:48 min der durchschnittlich am schnellsten bewertete Datensatz. Wie in [Abbildung 5.3](#) zu erkennen, handelt es sich hierbei um einen sehr einfach einzusehenden Datensatz, der bis auf den fehlenden Teil des Aortenbogens vollständig dargestellt wird. Er lässt sich zudem leicht in Perspektiven rotieren, aus denen entweder die Aorta oder die Pulmonalarterie nicht verdeckt werden. Da dieser Datensatz von einem Probanden stammt, findet sich hier nur eine leichte helikale Verwirbelung im Aortenbogen.

Im Gegensatz hierzu zeichnen sich die mit 2:40 und 2:41 min durchschnittlich am längsten untersuchten Datensätze *2_2012_Fallot* und *Aortenbypass_1_2011* durch besonders schlechte Sichtverhältnisse aus (siehe [Abbildung 5.1](#) sowie [Abbildung 5.2](#)). In beiden Fällen blockieren Pulmonalarterie sowie rechter Ventrikel den Blick auf die Aorta. Bei *Aortenbypass_1_2011* liegen Pulmonalarterie und Aorta zudem so dicht beieinander, dass sie in der Visualisierung miteinander verschmelzen.

Bei den Datensätzen *ProtheseISTA_11_2012* und *Flow_MD_3_2014*, deren durchschnittliche Bearbeitungsdauer ähnlich lang war, lagen die Probleme vor allem in der unvollständigen Darstellung (siehe [Abbildung 5.6](#) sowie [Abbildung 5.5](#)) der Anatomie. Bei letzterem kam zudem noch Verwirrung hinsichtlich des ausgeprägten Inflow-Jets auf. Da der Datensatz immer in der MIP-Darstellung angezeigt wurde, ordneten viele Teilnehmer (insbesondere Laien) den Inflow-Jet sogar fälschlicherweise der Pulmonalarterie zu.

Auch bei der in [Abbildung 5.8](#) dargestellten Untersuchungszeit pro Teilnehmer zeigen sich mitunter starke Unterschiede. So nahm sich Teilnehmer P3 beispielsweise besonders viel Zeit für die Bearbeitung der Datensätze, während P7 und P8 die Untersuchungen sehr schnell durchführten. Einen nur unwesentlichen Einfluss auf die Bearbeitungsdauer hat die Auswahl der Darstellungsart, wie oben links in [Abbildung 5.9](#) deutlich wird. Rampenfunktion und MIP haben eine identische durchschnittliche Bearbeitungsdauer, während die Bearbeitung der soliden Körper nur unwesentlich länger dauerte.

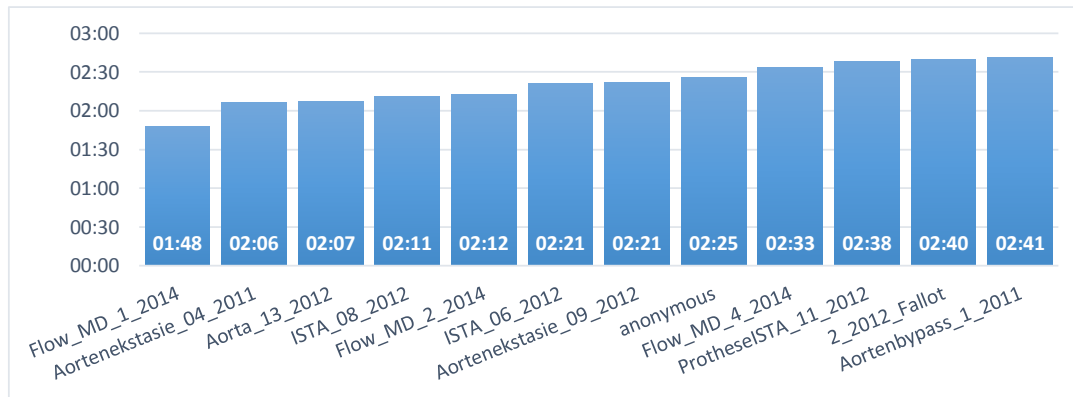


Abbildung 5.7: Durchschnittliche Bearbeitungsdauer in Minuten pro Datensatz, geordnet nach Zeit.

Einen etwas stärkeren Einfluss hat die Auswahl der weiteren Visualisierungsparameter. Die durchschnittliche Bearbeitungszeit gruppiert nach der Konfiguration des Prototypen ist ebenfalls in [Abbildung 5.8](#) aufgelistet. Die Kombination von aktivierten Overlays sowie der Verwendung von in Echtzeit berechneter λ_2 -Werte führt im Vergleich zu allen anderen Kombinationen zu einer erhöhten Bearbeitungsdauer. Einige der Teilnehmer gaben an, dass die Overlays Probleme bei der genauen Lokalisierung der Wirbel hervorriefen, insbesondere in Datensätzen mit starken Überlagerungen von Aorta und Pulmonalarterie. Zudem reagiert die Echtzeitberechnung von λ_2 -Werten sensibler auf Rauschen und führt daher zu einer erhöhten Anzahl von *false-positives* in laminarem Fluss. Deren Hervorhebung wird durch die Verwendung der Overlays noch verstärkt, was die erhöhte Bearbeitungsdauer erklären kann.

Die Probandendatensätze wurden, wie in [Abbildung 5.8](#) dargestellt, erwartungsgemäß am schnellsten bewertet. Überraschend ist hingegen, dass die leichten Datensätze die längste Bearbeitungszeit aufweisen, obwohl hier sowohl Anatomie als auch Verwirbelungen leicht erkennbar waren. Möglicherweise nahmen sich die Teilnehmer hier aufgrund der guten Sichtbarkeit der Gefäße besonders viel Zeit, diese zu untersuchen.

5.3.3 Ergebnisse der Wirbelfindung

Alle vier Datensatzgruppen bestanden aus jeweils drei Datensätzen, die von allen acht Teilnehmern untersucht wurden. Somit fanden insgesamt 24 Untersuchungen pro Datensatzgruppe statt.

Die Datensätze der Probanden-Gruppe wurden in den meisten Fällen korrekt als solche erkannt. In 13 von 24 Untersuchungen wurden die Datensätze als wirbelfrei beschrieben, in vier weiteren Fällen wurden lediglich die auch bei gesunden Personen vorhandenen leichten helikalen Wirbel im Aortenbogen markiert. Insgesamt wurde also in 17 von 24 Fällen korrekt bewertet. Viermal wurden Verwirbelungen markiert, die gegen Ende der Diastole aufgrund des abbremsenden Blutflusses auftraten und eigentlich keine relevanten Wirbelstrukturen darstellen. In drei Fällen wurden Flussstrukturen, die keine Verwirbelungen enthielten, fälschlicherweise als Wirbel erkannt. Dabei fällt auf, dass die falschen Erkennungen vor allem bei aktivierter Echtzeit- λ_2 -Berechnung auftraten (in 6 von 7 Fällen).

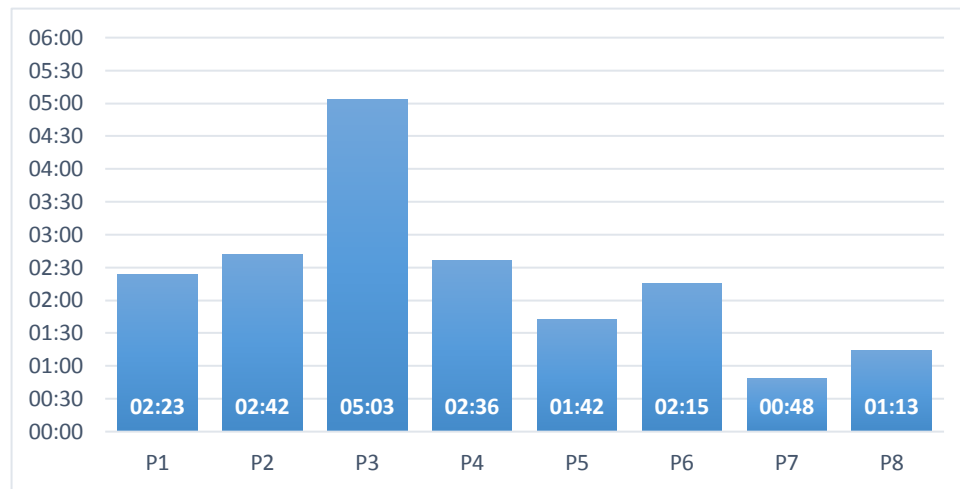


Abbildung 5.8: Durchschnittliche Bearbeitungsdauer in Minuten der Teilnehmer (P1 - P8).

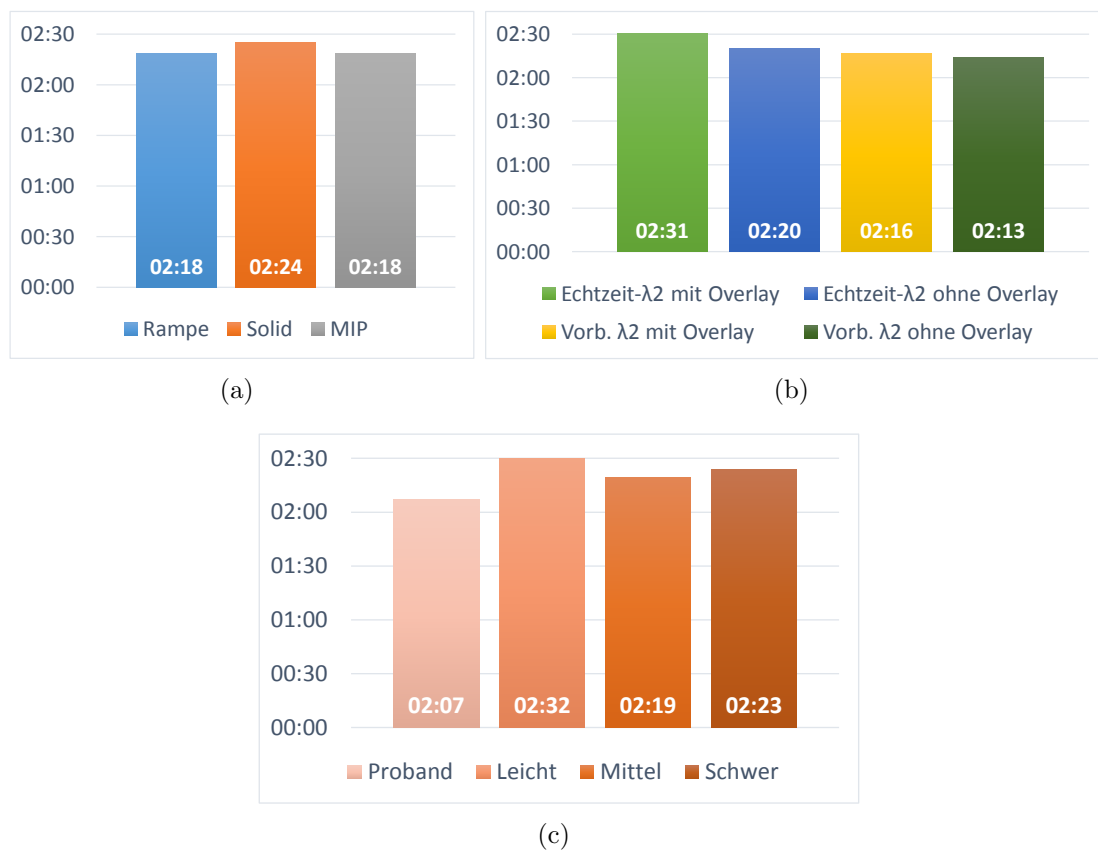


Abbildung 5.9: Durchschnittliche Bearbeitungsdauer in Minuten pro Darstellungsart (a), Konfiguration (b) sowie Datensatzgruppe (c).

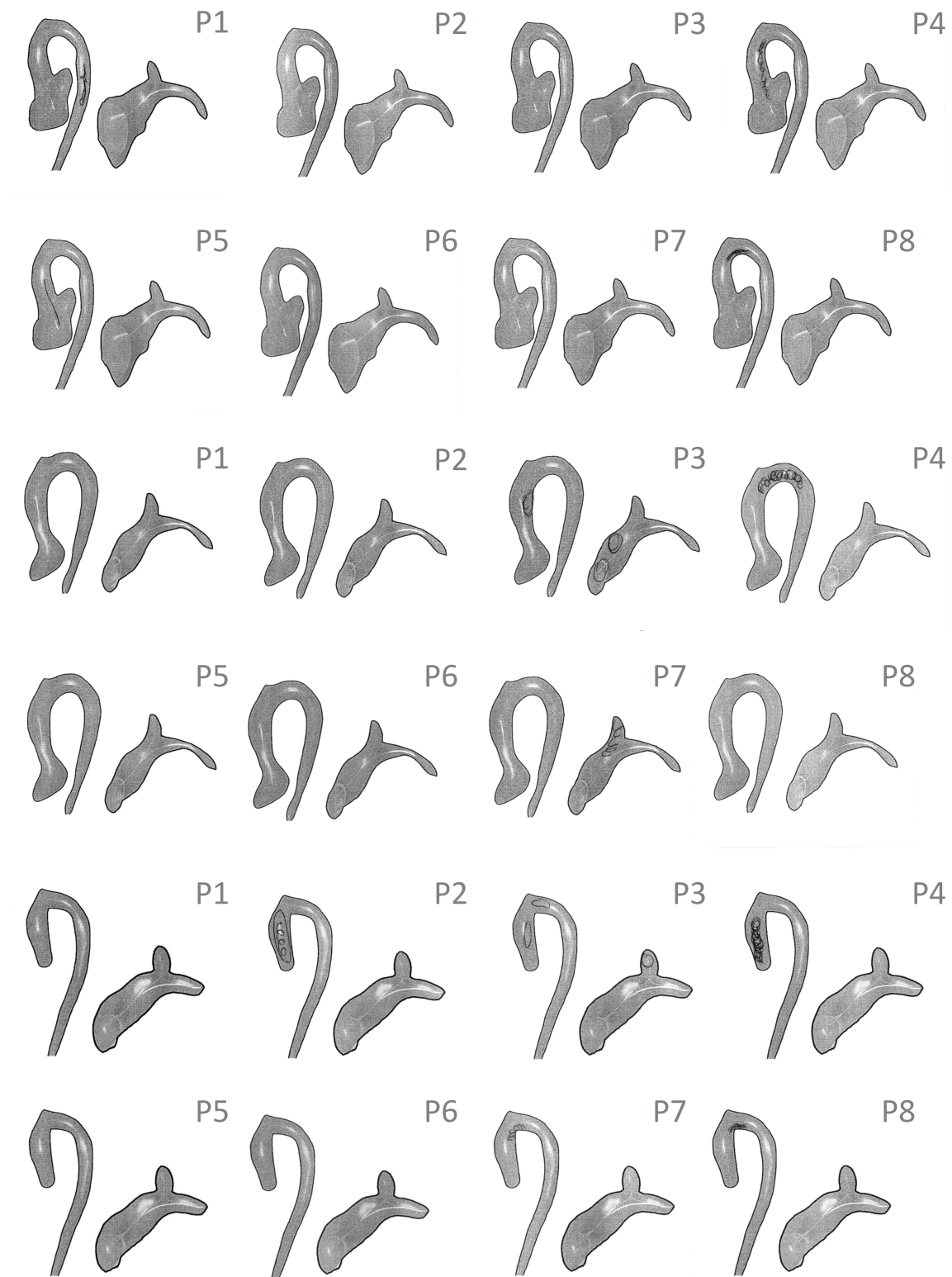


Abbildung 5.10: Skizzen der Teilnehmer zu den Probanden-Datensätzen *anonymous* (oben), *Aorta_13_2012* (Mitte) sowie *Flow_MD_1_2014* (unten).

2_2012_Fallot

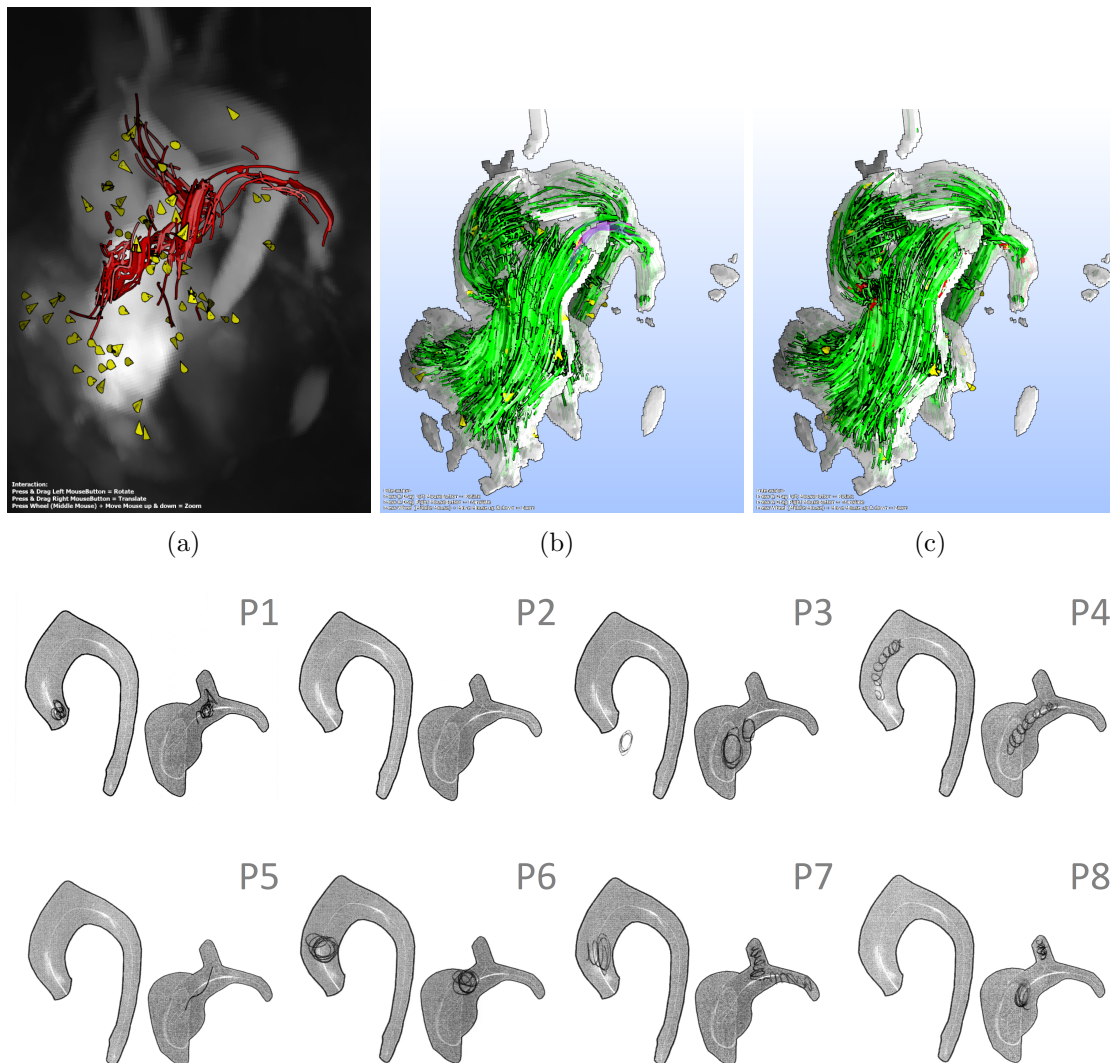


Abbildung 5.11: 2_2012_Fallot mit Darstellung der Verwirbelungen ohne laminaren Fluss (a), mit vorberechneten λ_2 -Werten und Overlays (b), in Echtzeit berechneten λ_2 -Werten ohne Overlays (c) sowie den Wirbelskizzen der Teilnehmer (unten).

Die Verwirbelungen in diesem Datensatz befinden sich im wesentlichen in der Pulmonalarterie. Sieben der acht Testpersonen haben diesen Wirbel erkannt, allerdings wurde dessen Größe oft falsch eingeschätzt. Zwei Testpersonen (P4 und P6), die diesen Datensatz beide ohne Overlays bearbeiteten, hatten Probleme, die Pulmonalarterie von der sehr eng anliegenden Aorta zu trennen und zeichneten daher einen Wirbel, der nur in der Pulmonalarterie auftrat, zusätzlich auch in der Aorta ein. Drei weitere Testpersonen (P1, P3 und P4), die Overlays verwendeten, zeichneten einen eigentlich im linken Ventrikel auftretenden Wirbel in den unteren Bereich der aufsteigenden Aorta ein. Einer von ihnen (P3) zeichnete diesen außerhalb der Aorta, da er erkannte, dass dieser Teil der Anatomie (linker Ventrikel) auf der Skizze nicht abgebildet war.

Aortenbypass_1_2011

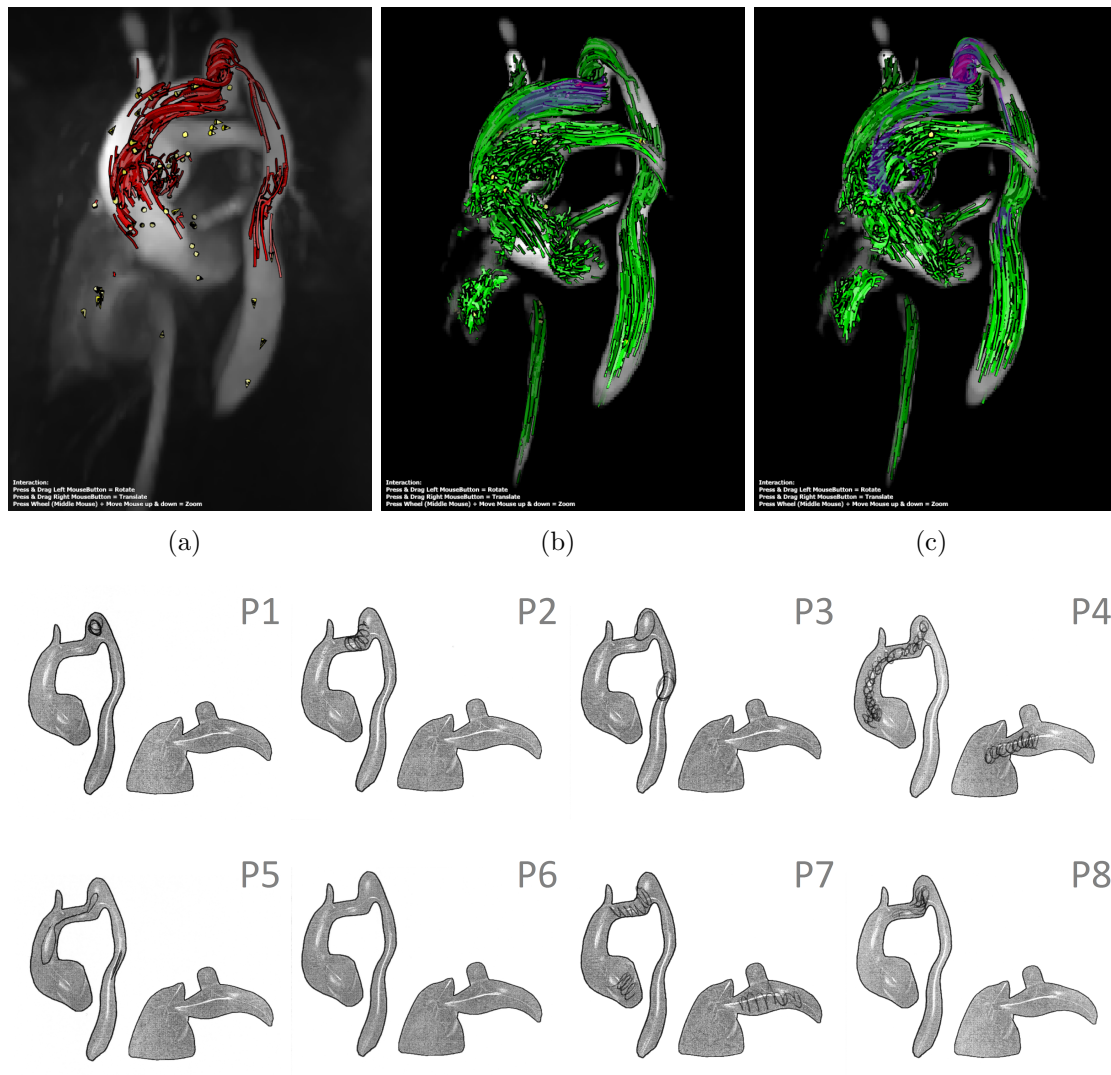


Abbildung 5.12: *Aortenbypass_1_2011* mit Darstellung der Verwirbelungen ohne laminaren Fluss (a), mit vorberechneten λ_2 -Werten (b), in Echtzeit berechneten λ_2 -Werten (c) sowie den Wirbelskizzen der Teilnehmer (unten).

Dieser Datensatz enthält einen großen Wirbel, der sich von der aufsteigenden Aorta bis in den Aortenbogen zieht. Wie im vorhergehenden Datensatz wurde dieser von sieben der acht Teilnehmer erkannt, aber in der Größe falsch eingeschätzt. Vier der Teilnehmer (P1, P2, P3, P8) zeichnen nur den stärksten Teil der Verwirbelung ein, der sich im Aortenbogen befindet, ein weiterer Teilnehmer (P7) erkannte zwar auch einen Wirbel in der aufsteigenden Aorta, zog jedoch keine Verbindung zum Aortenbogen. Nur zwei Teilnehmer (P4 und P5), bei denen Echtzeit- λ_2 -Werte visualisiert wurden, erkannten den vollständigen Wirbel. Eine weitere kleine Verwirbelung, die sich in der absteigenden Aorta befindet, wurde von zwei Teilnehmern (P3 und P5) erkannt, die Overlays verwendeten. Eine sehr schwache, helikale Verwirbelung in der Pulmonalarterie wurde von den zwei Teilnehmern P4 und P7 erkannt.

Aortenektasie_04_2011

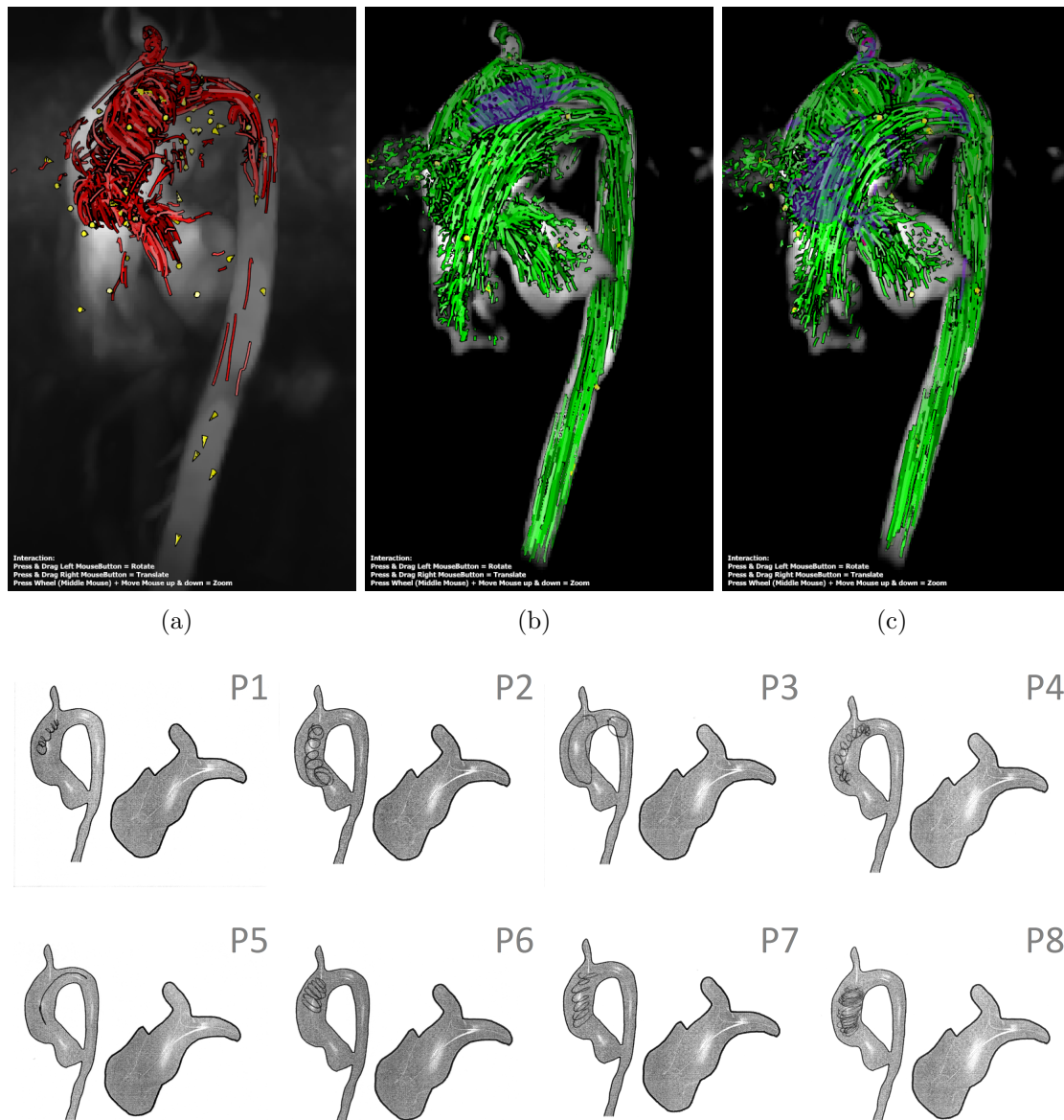


Abbildung 5.13: *Aortenektasie_04_2011* mit Darstellung der Verwirbelungen ohne laminaren Fluss (a), mit vorberechneten λ_2 -Werten (b), in Echtzeit berechneten λ_2 -Werten (c) sowie den Wirbelskizzen der Teilnehmer (unten).

Die Verwirbelung in diesem Datensatz reicht von der aufsteigenden Aorta bis in den absteigenden Aortenbogen und ist selbst ohne Wirbelextraktion durch pures Verfolgen der Flusslinien deutlich sichtbar. Dementsprechend wurde diese Verwirbelung auch von sämtlichen Teilnehmern erkannt und weitgehend korrekt lokalisiert. Lediglich einer der Teilnehmer (P3) wählte einen vergleichsweise hohen Schwellenwert für die Wirbelextraktion und trennte die Wirbelstruktur in zwei Teile, da die Verwirbelungen in einem kurzen Abschnitt des Aortenbogens schwächer ausgeprägt sind.

Aortenektasie_09_2012

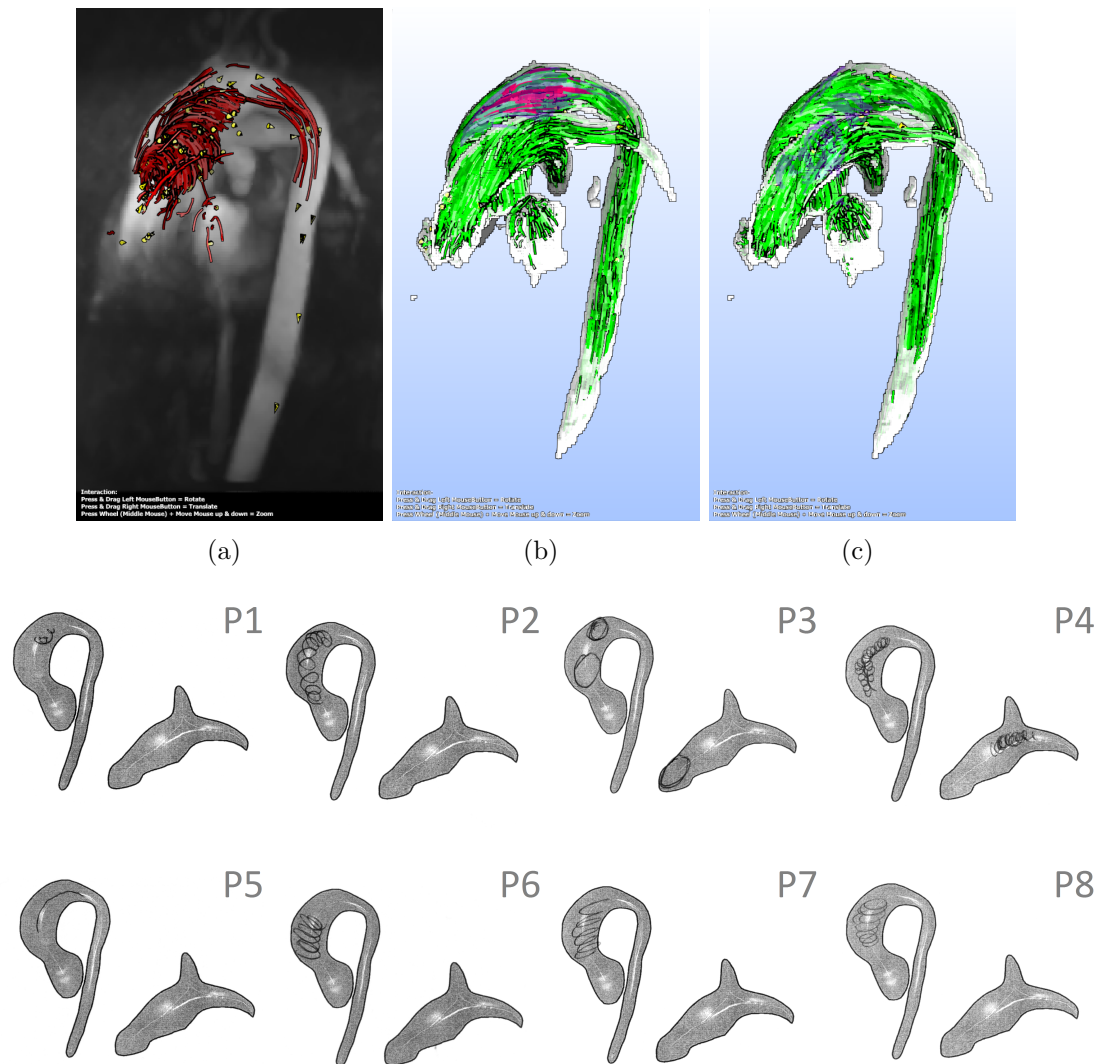


Abbildung 5.14: *Aortenektasie_09_2012* mit Darstellung der Verwirbelungen ohne laminaren Fluss (a), mit vorberechneten λ_2 -Werten (b), in Echtzeit berechneten λ_2 -Werten (c) sowie den Wirbelskizzen der Teilnehmer (unten).

Die Wirbelstruktur dieses Datensatzes ist der vorhergehenden sehr ähnlich und wurde dementsprechend erneut von allen Teilnehmern erkannt. Auch hier wurde von Teilnehmer P3 ein hoher Schwellenwert für die Wirbelextraktion gewählt, wodurch die eigentlich zusammengehörende Wirbelstruktur in zwei Teile aufgespalten wurde. Interessanterweise untersuchte dieser Teilnehmer den Datensatz auch mit einem extrem niedrigem Schwellenwert, sodass er als einziger eine kaum sichtbare Verwirbelung direkt hinter der Pulmonalklappe markierte. Ein weiterer Teilnehmer (P4) zeichnete einen Wirbel in der Pulmonalarterie ein, allerdings an einer Position, die keine nennenswerten Verwirbelungen enthält. Hier ist zu vermuten, dass der Teilnehmer Probleme bei der Unterscheidung zwischen Aorta und Pulmonalarterie hatte und demzufolge fälschlicherweise einen Wirbel der Aorta markierte.

Flow_MD_2_2014

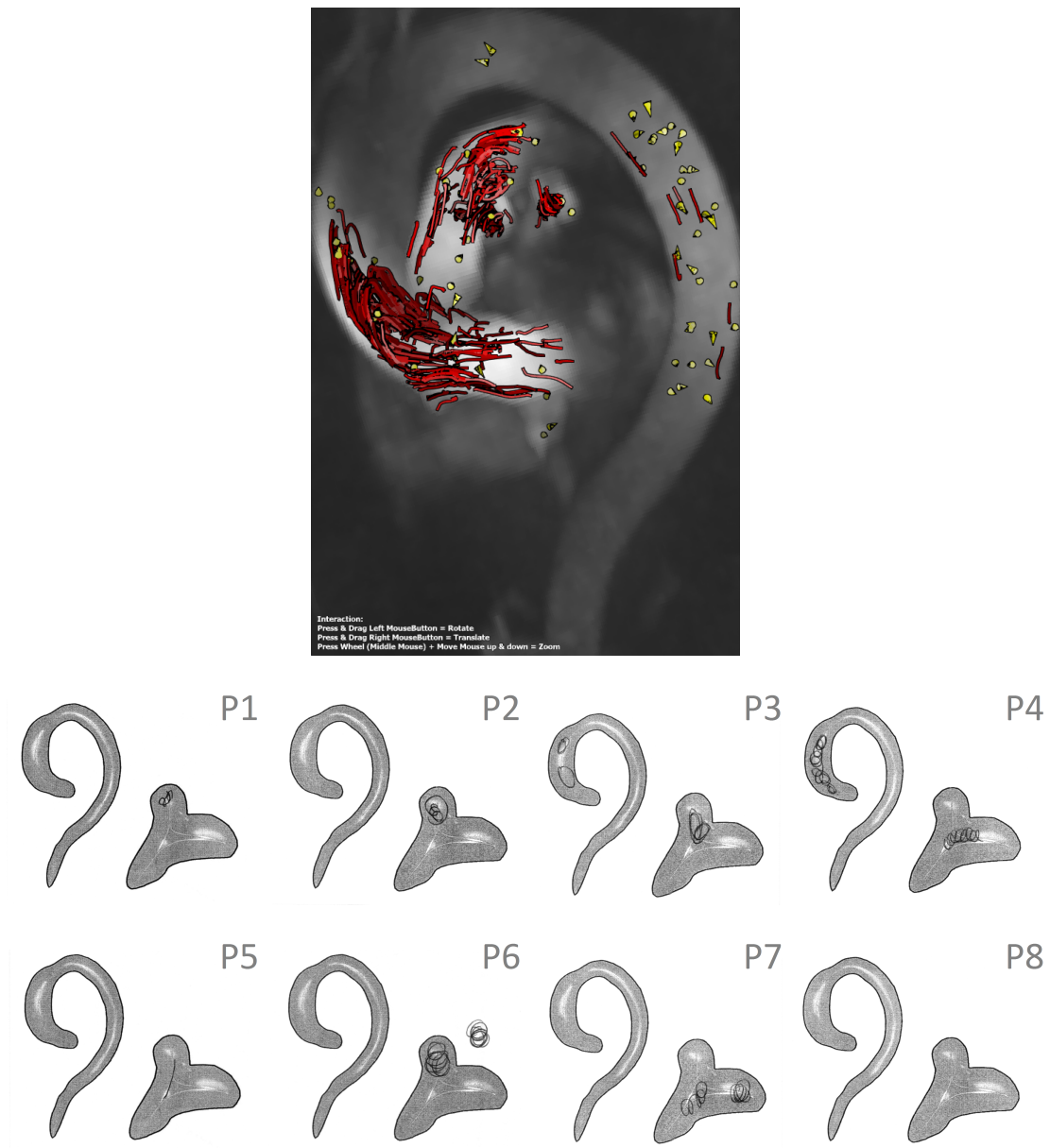


Abbildung 5.15: *Flow_MD_2_2014* mit Darstellung der Verwirbelungen ohne laminaren Fluss (oben) sowie den Wirbelskizzen der Teilnehmer (unten).

Aufgrund der bereits beschriebenen Probleme mit diesem Datensatz (siehe [Abschnitt 5.2](#)) ist eine Flussuntersuchung hier nur sehr eingeschränkt möglich, da lediglich Teile von Aorta und Pulmonalarterie sichtbar sind. Die in der Pulmonalarterie vorhandenen Verwirbelungen wurden von sieben der acht Teilnehmer erkannt, wobei es jedoch Schwierigkeiten bei der genauen Lokalisierung aufgrund der unvollständigen Visualisierung gab. Zwei der Teilnehmer (P2 und P4) markierten zudem den starken Inflow-Jet in der Aorta als Verwirbelung.

Flow_MD_4_2014

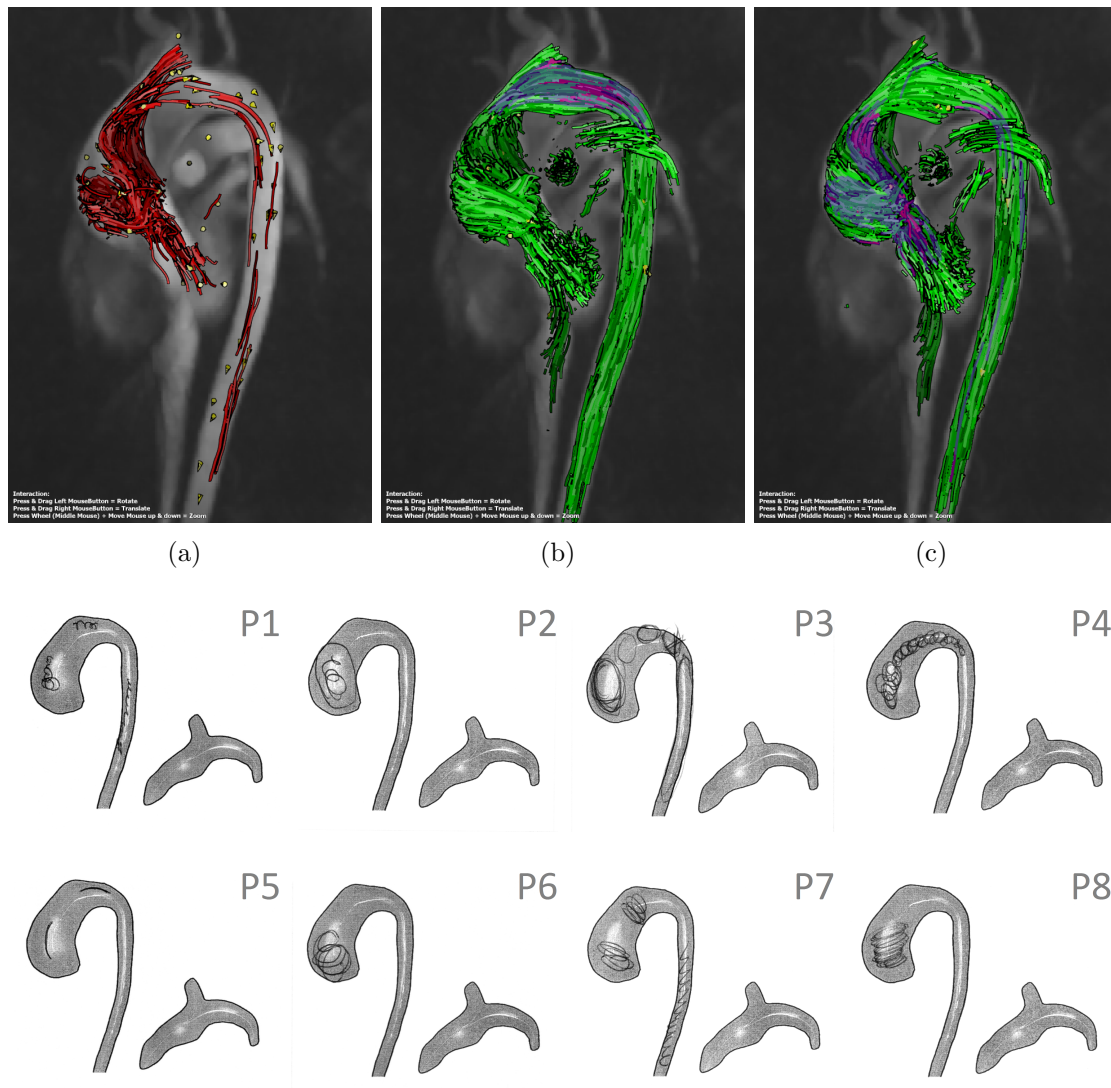


Abbildung 5.16: *Flow_MD_4_2014* mit Darstellung der Verwirbelungen ohne laminaren Fluss (a), mit vorberechneten λ_2 -Werten (b), in Echtzeit berechneten λ_2 -Werten (c) sowie den Wirbelskizzen der Teilnehmer (unten).

Auch dieser Datensatz zeichnet sich durch einen exzentrischen Inflow-Jet aus, wird aber wesentlich vollständiger dargestellt als der letzte. Dieser Inflow-Jet verursacht eine deutlich sichtbare Verwirbelung in der aufsteigenden Aorta sowie eine weitere im Aortenbogen. Die Verwirbelung in der aufsteigenden Aorta wurde von allen Teilnehmern erkannt, jedoch markierten nur fünf der acht Teilnehmer die zweite Verwirbelung im Aortenbogen. In drei Fällen (P1, P3 und P7) wurde zudem ein weiterer, weniger deutlicher Wirbel in der absteigenden Aorta markiert.

ISTA_06_2012

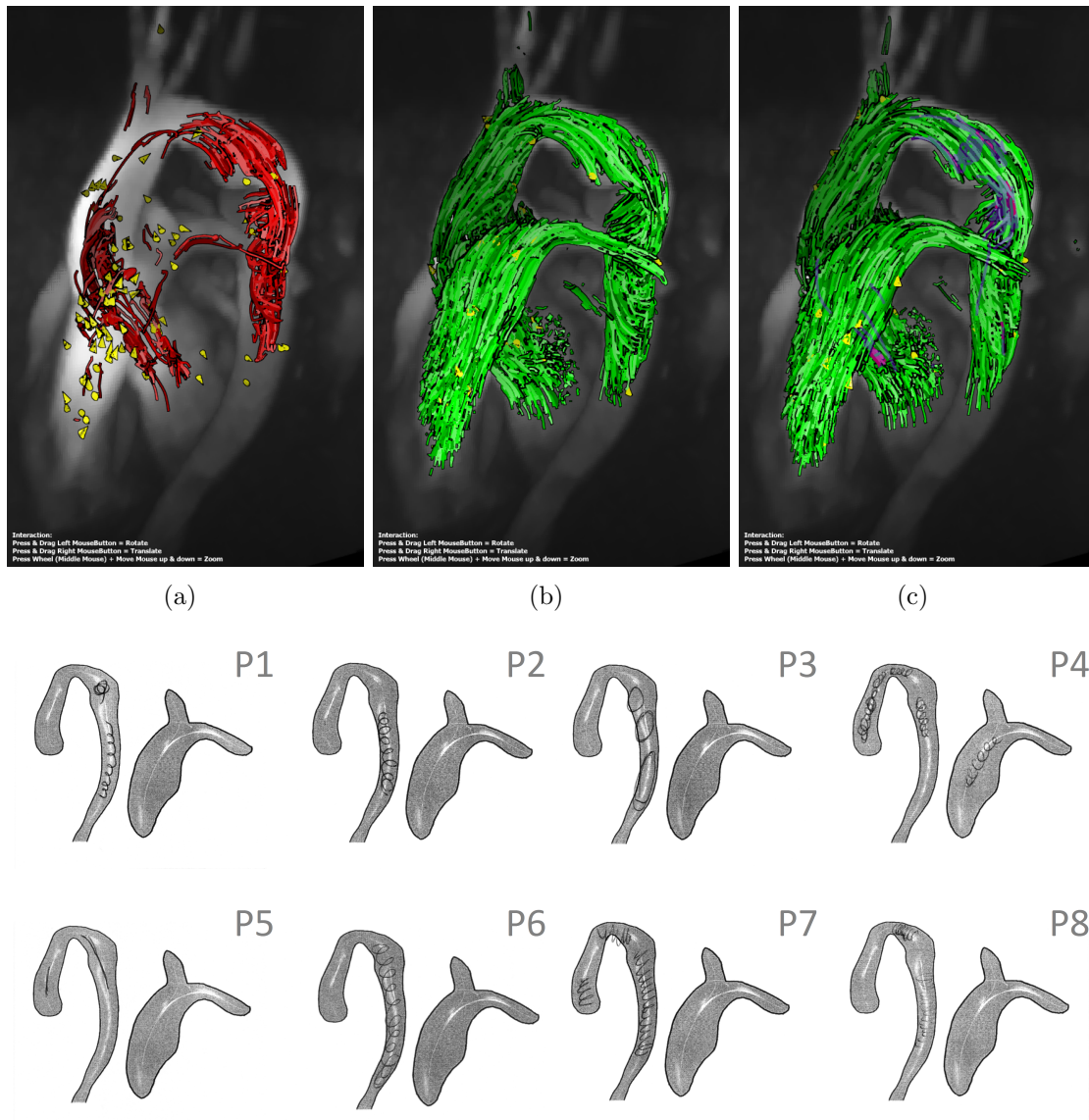


Abbildung 5.17: *ISTA_06_2012* mit Darstellung der Verwirbelungen ohne laminaren Fluss (a), mit vorberechneten λ_2 -Werten (b), in Echtzeit berechneten λ_2 -Werten (c) sowie den Wirbelskizzen der Teilnehmer (unten).

Dieser Datensatz enthält Verwirbelungen sowohl in der auf-, als auch in der absteigenden Aorta. Die Verwirbelungen in der absteigenden Aorta sind dabei schon durch bloßes Verfolgen der Pfadlinien gut erkennbar und wurden demzufolge von allen Teilnehmern zuverlässig erkannt. Vier der Teilnehmer (P1, P3, P7 und P8) teilten die Wirbelstruktur jedoch in mehrere Segmente auf. Die in der aufsteigenden Aorta vorhandenen Wirbel sind ohne Overlays oder starke Anpassung des λ_2 -Schwellenwertes jedoch kaum zu erkennen und wurden nur von drei Teilnehmern (P4, P5 und P7) markiert. Einer dieser Teilnehmer (P4) zeichnete zudem eine nicht existierende Verwirbelung in der Pulmonalarterie ein.

ISTA_08_2012

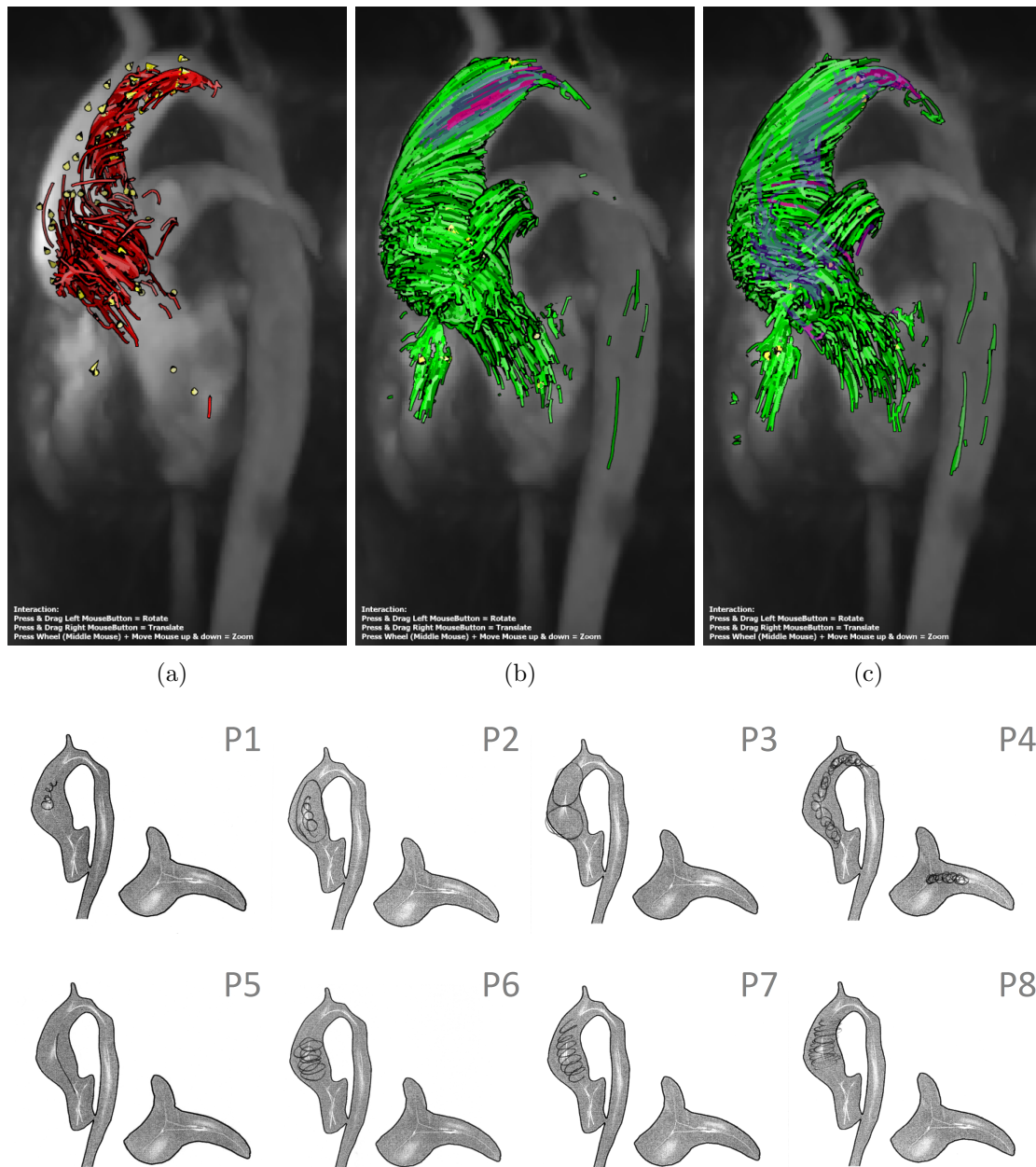


Abbildung 5.18: *ISTA_08_2012* mit Darstellung der Verwirbelungen ohne laminaren Fluss (a), mit vorberechneten λ_2 -Werten (b), in Echtzeit berechneten λ_2 -Werten (c) sowie den Wirbelskizzen der Teilnehmer (unten).

Auch dieser Datensatz enthält einen gut sichtbaren Wirbel in der aufsteigenden Aorta. Dieser wurde von allen Teilnehmern erkannt, wobei die Größe des Wirbels in zwei Fällen (P1 und P6) unterschätzt wurde. Einer der Teilnehmer (P4) markierte einen Wirbel in der Pulmonalarterie, die jedoch keine relevanten Wirbelstrukturen enthält. Da Aorta und Pulmonalarterie in diesem Datensatz sehr dicht beieinander liegen und deren Differenzierung zusätzlich durch die MIP-Darstellungsart erschwert wird, handelt es sich hierbei höchstwahrscheinlich um einen aus der Aorta stammenden Wirbel, der falsch eingetragen wurde.

ProtheseISTA_11_2012

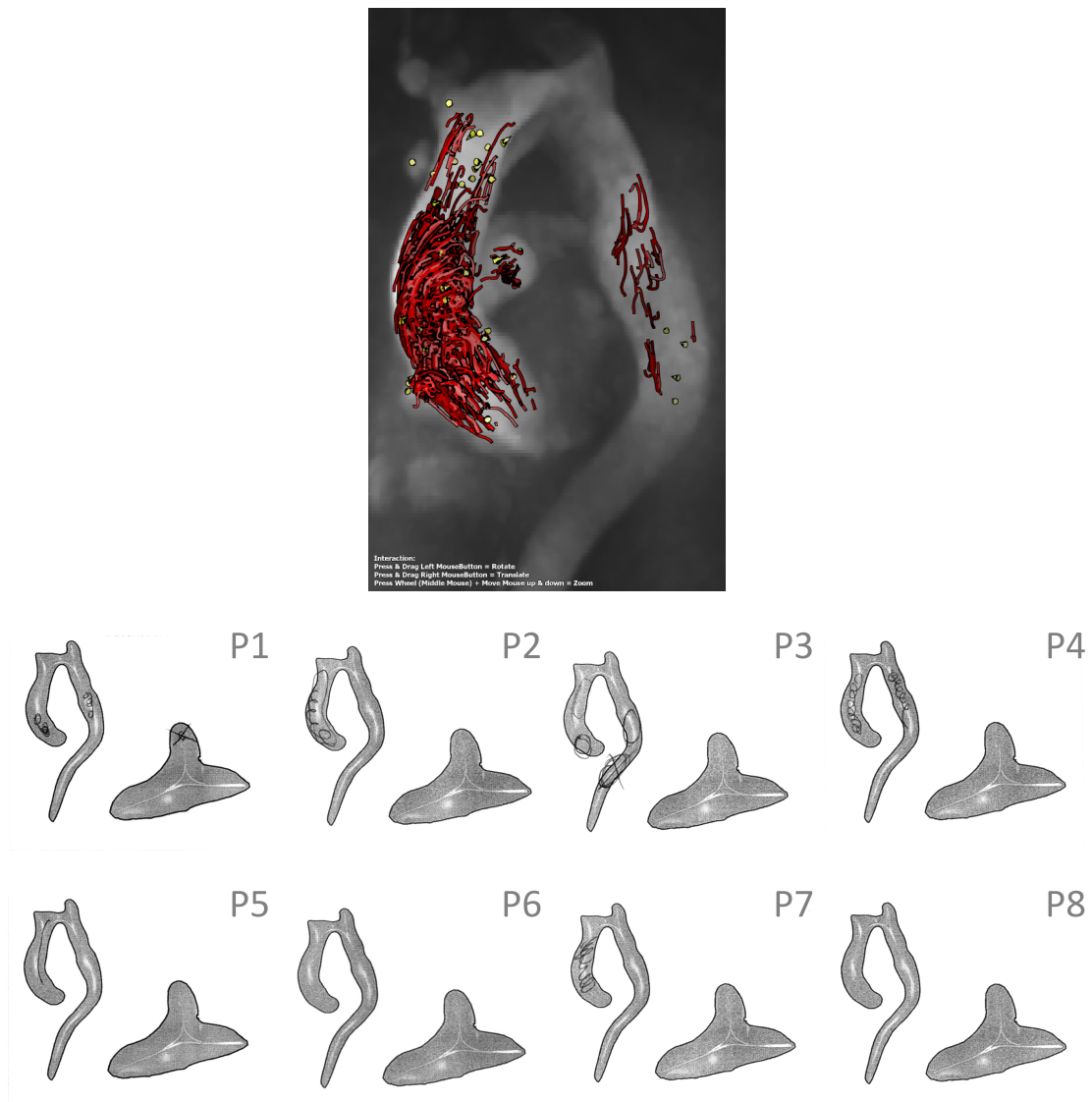


Abbildung 5.19: *ProtheseISTA_11_2012* mit Darstellung der Verwirbelungen ohne laminaren Fluss (oben) sowie den Wirbelskizzen der Teilnehmer (unten).

Wie bereits in [Abschnitt 5.2](#) beschrieben, wird dieser Datensatz nur unvollständig visualisiert und ermöglicht daher nur eine eingeschränkte Begutachtung des Flusses. Gut sichtbar ist hier ein Wirbel in der aufsteigenden Aorta, welcher von sechs der acht Teilnehmer erkannt wurde. Drei Teilnehmer (P1, P4 und P5) markierten zudem eine Verwirbelung in der absteigenden Aorta, die sich allerdings erst gegen Ende der Diastole durch das Abbremsen des Blutes herausbildet und damit keine relevante Wirbelstruktur darstellt. Der erste Teilnehmer markierte zudem eine sehr schwache Verwirbelung in der Pulmonalarterie, die aufgrund der Nähe zur Aorta nur schwer zu erkennen war.

In den meisten Fällen konnten die Teilnehmer größere Wirbelstrukturen in der Aorta problemlos finden. Insgesamt wurden 72 Untersuchungen an Patientendatensätzen durchgeführt, in 53 davon (73.6%) wurden alle medizinisch relevanten Wirbelstrukturen in den Datensätzen durch die Teilnehmer markiert. Bei der Bewertung der Größe und Position dieser Strukturen kam es jedoch zu Abweichungen, da deren Hervorhebung wesentlich von dem gewählten λ_2 -Schwellenwert abhängt. Interessanterweise hat die Verwendung von Overlays und in Echtzeit berechneten λ_2 -Werten nur geringen Einfluss auf die Qualität der Ergebnisse. So kompensierten die Teilnehmer das Fehlen von Overlays durch eine Verringerung des λ_2 -Schwellenwertes, sodass auch innen liegende Verwirbelungen besser sichtbar wurden.

Bei Datensätzen, welche durch die automatische Histogrammanalyse nur sehr unvollständig visualisiert werden konnten, war eine vollständige Bewertung des Blutflusses nicht möglich. Da hier der Fluss in Teilen der Anatomie nicht visualisiert wurde, waren eventuell in diesen Bereichen vorhandene Wirbel damit nicht erkennbar.

Da es sich bei den Teilnehmern nicht um Mediziner handelte, gab es des Öfteren Unstimmigkeiten darüber, ab welcher Größe bzw. Stärke eine Verwirbelung im Fluss als Wirbel zu klassifizieren ist. So markierten einige Teilnehmer beispielsweise abbremsenden Fluss am Ende der Systole als Verwirbelungen. In insgesamt sieben der 72 Untersuchungen (9.7%) wurden derartige Strukturen als Wirbel markiert. Insgesamt acht mal (11.1%) wurden Strukturen fälschlicherweise als Verwirbelungen gekennzeichnet. Der Hauptgrund hierfür war, dass die Teilnehmer in bestimmten Situationen Aorta und Pulmonalarterie nicht korrekt voneinander trennen konnten.

5.3.4 Befragungsergebnisse

Nach der Auswertung der 13 Datensätze wurden den Teilnehmern Fragen zur Handhabung des Prototypen sowie der Visualisierung der Datensätze auf einer Likert-Skala von 0 (negativ) bis 4 (positiv) gestellt. Die durchschnittlichen Bewertungen sind in [Abbildung 5.21](#) dargestellt.

Der Überblick über die Datensätze wurde mit 3 durchschnittlich als gut bewertet, wobei sich mehrere Teilnehmer eine Funktion zum Ausblenden der Aorta oder Pulmonalarterie zur Reduzierung von Verdeckungen wünschten.

Häufig wurde der automatisch gesetzte λ_2 -Schwellenwert bemängelt, da dieser für die meisten Datensätze zu hoch eingestellt war. Häufig mussten Teilnehmer den Wert senken, um alle Wirbel vollständig angezeigt zu bekommen. Die Abhängigkeit der Qualität der Visualisierung vom gewählten Datensatz wurde ebenfalls häufig negativ angemerkt, da einige Datensätze aufgrund von niedrigem Kontrast nur sehr unvollständig angezeigt wurden. Insgesamt wurde die automatische Visualisierung mit 2.38 nur durchschnittlich bewertet.

Die Möglichkeit zur Änderung der Transferfunktion mithilfe eines Sliders sowie die Einstellbarkeit der Animationsgeschwindigkeit wurden als sehr positiv bewertet. Von einigen Teilnehmern wurde gewünscht, zusätzlich die Anzahl oder Dicke der Pfadlinien reduzieren zu können. Diese Funktion existiert, war jedoch in der für den Test verwendeten Version des Prototypen nicht freigeschaltet. Insgesamt erreichte die Anpassbarkeit der Visualisierung eine durchschnittliche Bewertung von 3.5. Hier

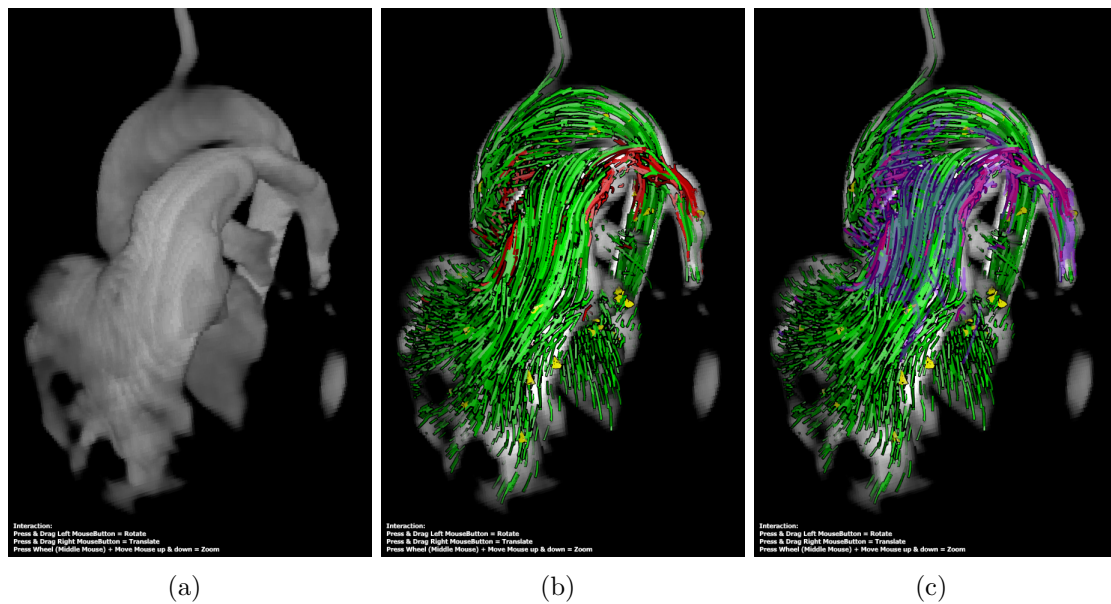


Abbildung 5.20: Verdeckungen im Datensatz *2_2012_Fallot* (a). Ohne Overlays sind die Verdeckungsverhältnisse leicht erkennbar (b), bei aktivierten Overlays ist die Zuordnung der Wirbel zu Aorta oder Pulmonalarterie wesentlich schwieriger (c).

zeigt sich, dass automatisch gesetzte Einstellungen nur als ein Vorschlag dienen und den Anwendern weiterhin die Möglichkeit gegeben werden sollte, diese anzupassen.

Die gelb dargestellten Partikel wurden von vielen Teilnehmern erst durch Nachfragen als Flusspfeile erkannt. Zudem war es häufig nur sehr eingeschränkt möglich, den Flussverlauf im Inneren des Gefäßes zu verfolgen, wenn dieser nicht durch Overlays hervorgehoben wurde. Andererseits wurden Overlays manchmal auch als verwirrend empfunden, wenn sich anatomische Strukturen überlagerten. Ein Beispiel hierfür ist in [Abbildung 5.20](#) dargestellt. In diesen Fällen war es durch die verfälschten Tiefeninformationen schwierig, den zugehörigen Wirbel einem bestimmten Gefäß zuzuordnen. Insgesamt wurde die Nachvollziehbarkeit des Blutflusses anhand der Pfadlinien mit 2.8 bewertet.

Die Farbgebung der Pfadlinien (grün für laminare Strukturen, rot für Verwirbelungen) war für die meisten Teilnehmer intuitiv verständlich und wurde demzufolge mit 3.63 sehr gut bewertet. Zudem wurde positiv hervorgehoben, dass diese sich bei Rampenfunktion und MIP deutlich vom dunklen Hintergrund abheben. Für Irritationen sorgten hingegen gelegentlich die blauen Overlays. Da diese sich semitransparent über die Partikel legen, erschienen Verwirbelungen bei aktivierten Overlays nicht rot, sondern lila. Mit 3.25 und 3 ebenfalls gut bewertet wurde die Erfassung sämtlicher Gefäßbereiche durch die Pfadlinien sowie die Möglichkeit, den Fluss vollständig auszublenden und so die Anatomie separat betrachten zu können.

Zudem wurde die Erkennbarkeit der Anatomie, Räumlichkeit der Darstellung sowie Sichtbarkeit der Partikel in den drei Darstellungsarten bewertet. Die Ergebnisse sind in [Abbildung 5.22](#) dargestellt.

Die Erkennbarkeit der Anatomie wurde für alle drei Darstellungsarten positiv bewertet. Hierzu muss erwähnt werden, dass die Teilnehmer nur die Visualisierung selbst,

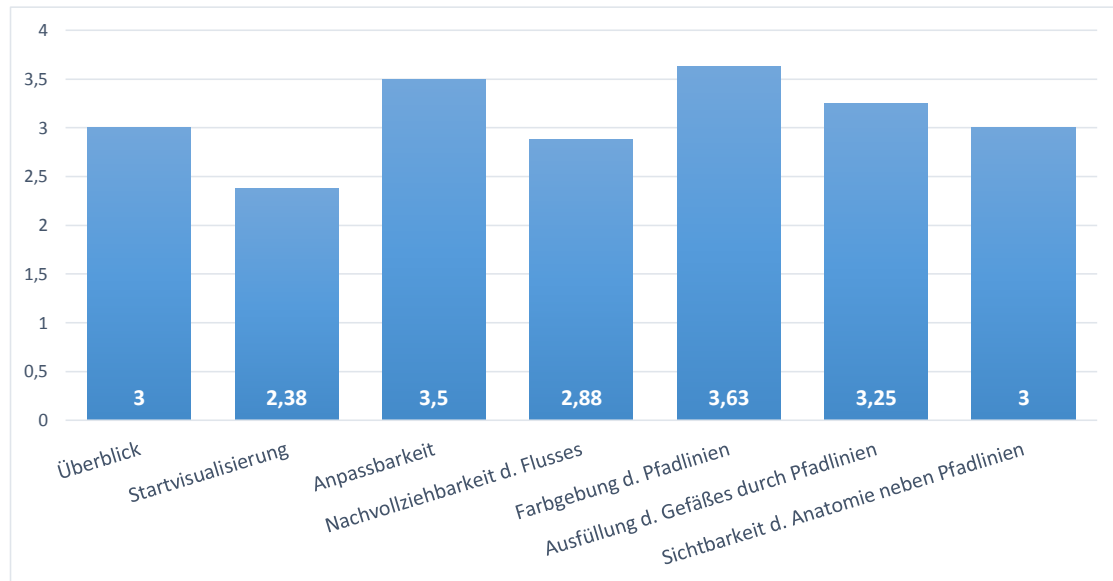


Abbildung 5.21: Allgemeine Bewertung des Prototypen sowie der Visualisierung auf einer Skala von 0 (negativ) bis 4 (positiv).

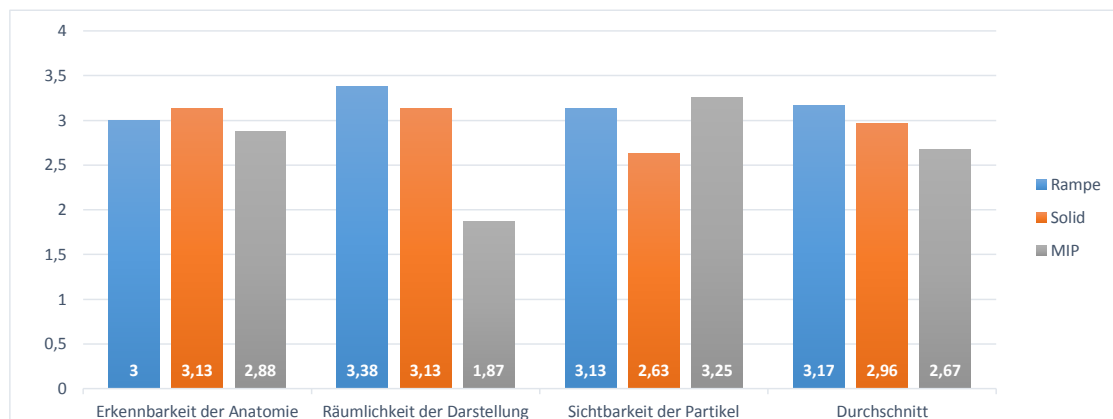


Abbildung 5.22: Bewertung der einzelnen Darstellungsarten auf einer Skala von 0 bis 4.

nicht die Datensätze bewerten sollten. Die soliden Körper schneiden hier minimal besser ab, da sie durch die Silhouetten am stärksten vom Hintergrund abgegrenzt sind. Die MIP wurde dementsprechend schlechter bewertet. Jedoch wurde positiv angemerkt, dass hier die Anatomie in jedem Fall vollständig dargestellt wird und auch kontrastarme Strukturen erkennbar bleiben.

Bezüglich der Räumlichkeit der Darstellung erreicht die erweiterte Rampenfunktion das beste Ergebnis. Dies ist überraschend, da die Plastizität durch fehlende Beleuchtung geringer ist als die der soliden Körper. Die MIP erreicht aufgrund der verworfenen Tiefeninformationen das schlechteste Ergebnis. Bei Überlappungen hatten viele Teilnehmer Probleme, die entsprechenden Strukturen zu differenzieren. In einem solchen Fall wurde versucht, durch Rotieren des Datensatzes eine bessere Übersicht zu erhalten. Mehrere Teilnehmer berichteten, dass sich dieses Problem durch die Partikelvisualisierung abmildern lies, da diese Tiefenverhältnisse berücksichtigt.

Die Darstellung der Partikel wurde bei der MIP am positivsten bewertet, da Partikel hier nicht von der Anatomie verdeckt werden können. Einige Teilnehmer fanden es störend, dass bei der soliden Körper-Visualisierung sowohl bei den Partikeln als auch der Anatomie Silhouetten eingesetzt wurden. Dies könnte erklären, warum diese bei der Bewertung der Sichtbarkeit der Partikel schlechter bewertet wurden.

Wie in [Abbildung 5.23](#) zu sehen, ist die Präferenz für eine der drei Darstellungsmethoden äußerst subjektiv. Obwohl die MIP beispielsweise durchschnittlich schlechter bewertet wurde als die anderen Verfahren, haben vier der acht Teilnehmer sie als beste oder eine der besten Darstellungsarten gewählt. Die Rampenfunktion hingegen hat durchschnittlich die besten Bewertungen erhalten, wurde jedoch nur von drei Teilnehmern als beste oder eine der besten Darstellungsarten gewählt. Die soliden Körper wurden sogar von fünf Teilnehmern als beste Darstellungsart ausgewählt.

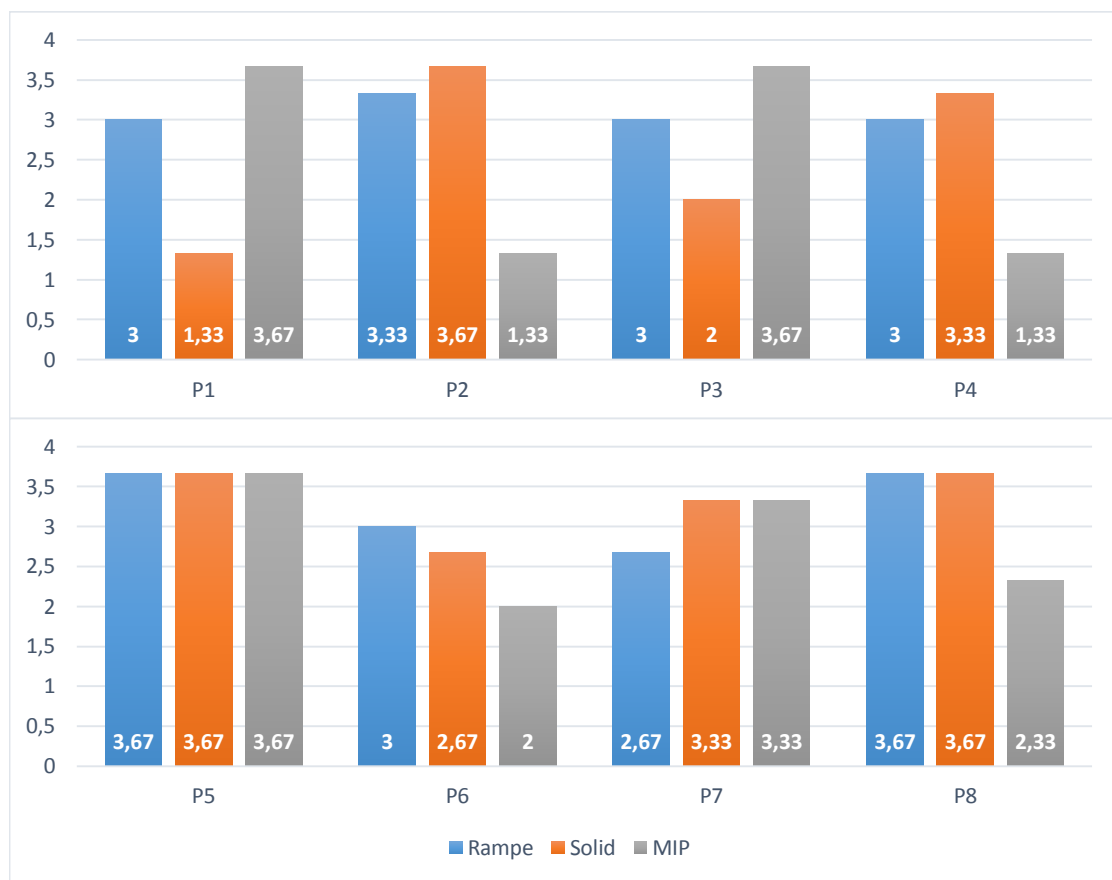


Abbildung 5.23: Durchschnittliche Bewertung der Darstellungsarten pro Anwender auf einer Skala von 0 bis 4.

5.4 Performance

Ein Ziel dieser Arbeit war, eine echtzeitfähige Methode zur Visualisierung von Flussdaten zu entwickeln. Aus diesem Grund ist eine Untersuchung der Performance des Prototyps von großer Bedeutung. Diese wurde auf einem System mit *Intel Core i7-4790K* CPU durchgeführt, der über 4 mit 4 GHz getaktete Kerne verfügt und somit

durch *Hyper Threading* bis zu 8 Threads gleichzeitig ausführen kann. Des Weiteren sind 16 GB Dual-DDR3-RAM sowie eine *nVidia GeForce GTX 980* mit 4 GB GDDR5-Speicher installiert.

In [Tabelle 5.2](#) werden die Anzahl der Frames pro Sekunde (*FPS*) exemplarisch für drei Datensätze dargestellt. Zusätzlich zu den bereits vorgestellten Darstellungsarten MIP, solider Körper und Rampenfunktion wurde hier eine vierte Variante aufgenommen, bei der ausschließlich die Partikel gerendert werden. Das Gefäß ist somit nicht sichtbar. Erwartungsgemäß wird mit dieser Einstellung die höchste Framerate erreicht. Rampenfunktion und solide Körper haben von allen Darstellungsarten den höchsten Berechnungsaufwand. Die Auflösung des Datensatzes hingegen spielt nur eine untergeordnete Rolle, da diese keinen Einfluss auf die Anzahl der Sample-Punkte auf den Strahlen der DVR-Berechnung hat. Obwohl sich die drei in [Tabelle 5.2](#) aufgeführten Datensätze in ihrer Auflösung in z-Richtung (15, 20 und 25) unterscheiden, ist kein signifikanter Unterschied in der Framerate zu erkennen.

Betrachtet man nur die Visualisierung der Anatomie durch DVR, so ergibt sich wie in [Tabelle 5.3](#) dargestellt, ein ähnliches Bild. Auch hier ist die Framerate offensichtlich unabhängig vom gewählten Datensatz und die Darstellung mittels MIP ist etwas schneller als die der anderen Varianten.

Datensatz	Nur Partikel	MIP	Solider Körper	Rampenfunktion
Aortenbypass_1_2011	333	56	50 / 43	50 / 43
Flow_MD_1_2014	333	56	53 / 43	53 / 45
Aortenektasie_04_2011	333	56	50 / 43	50 / 43

Tabelle 5.2: FPS für unterschiedliche Datensätze und Darstellungsarten bei aktivierter Flussvisualisierung mit 4096 Partikeln. Für solide Körper und Rampenfunktion ist zusätzlich die FPS-Zahl bei aktiviertem Frontface-Culling angegeben. Die Auflösung des Rendering-Bereichs betrug 828 x 918 Pixel.

Datensatz	MIP	Solider Körper	Rampenfunktion
Aortenbypass_1_2011	250	200	200
Flow_MD_1_2014	250	200	200
Aortenektasie_04_2011	250	200	200

Tabelle 5.3: FPS für unterschiedliche Datensätze und Darstellungsarten ohne Partikelvisualisierung. Die Auflösung des Rendering-Bereichs betrug 828 x 918 Pixel.

Da das GPU-basierte Raycasting lediglich für Pixel ausgeführt wird, auf welchen der Datensatz abgebildet wird, ist die Framerate stark abhängig von dessen Rotation und Zoom. In [Abbildung 5.24](#) wird der Einfluss unterschiedlicher Betrachtungswinkel und -positionen dargestellt. Je höher die Anzahl der Pixel ist, auf denen die Anatomie abgebildet wird, desto geringer ist die Framerate.

Interessanterweise wirken sich die Verwendung von RK4 zur Integration der Partikel sowie die Echtzeit- λ_2 -Berechnung selbst bei einer Erhöhung der Partikelzahl auf 8192 nicht auf die Framerate aus. Insgesamt zeigt sich, dass sämtliche Darstellungsarten unabhängig von weiteren Einstellungen und verwendetem Datensatz flüssig dargestellt werden können. Die geringste gemessene Framerate beträgt 43 Frames pro Sekunde.

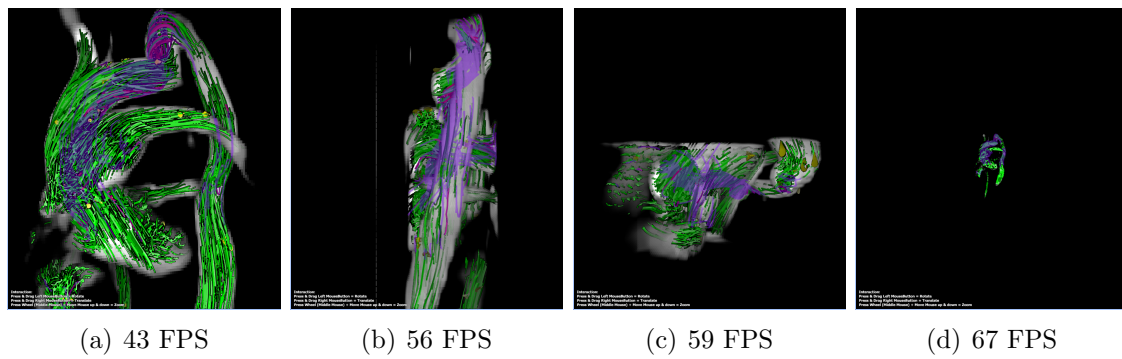


Abbildung 5.24: Unterschiedliche Blickwinkel und Skalierungen und ihre Auswirkungen auf die Framerate.

Der begrenzende Faktor in dieser Umgebung war eindeutig die Leistung der Grafikkarte, da diese den Hauptteil der Arbeit verrichtet. Die CPU kommt nur beim Laden der Datensätze verstärkt zum Einsatz, da sie die Daten auf den Grafikspeicher kopiert sowie einige Filterungen (z.B. $tStDev$) vornimmt. Während beim Laden der Datensätze die CPU vollständig ausgelastet wird, steigt die Auslastung bei der Visualisierung unabhängig von der Auflösung maximal auf 12.5%. Die Tatsache, dass es sich hierbei um genau $1/8$ der CPU-Leistung und damit um die Leistung eines einzelnen logischen Kernels handelt, ist Zufall. Auch während der Visualisierung wird die Last auf alle Kerne verteilt. Ein leistungsfähigerer Prozessor hätte somit lediglich Einfluss auf die Ladezeit der Datensätze, nicht jedoch auf die Framerate der Visualisierung.

6. Zusammenfassung und Ausblick

Das Ziel dieser Arbeit war die Entwicklung und Evaluation einer Methode zur qualitativen Analyse von kardialen 4D-PC-MRI-Daten, die durch Automatisierung und Hervorhebung von interessanten Flusscharakteristika eine zeitsparende Exploration des Blutflusses erlaubt, ohne dass eine Segmentierung der Daten erforderlich ist.

Die Echtzeitfähigkeit dieses Ansatzes wurde durch den Einsatz von GPU-Raycasting zur Darstellung der Anatomie sowie eines GPU-gestützten Partikelsystems für die Flussvisualisierung erreicht. Durch eine Histogrammanalyse werden Visualisierungsparameter wie Schwellenwerte oder Transferfunktionen automatisch bestimmt, sodass eine manuelle Segmentierung nicht erforderlich ist. Durch Berechnungen bedingte Wartezeiten treten lediglich beim Laden eines Datensatzes auf und dauern nur einige Sekunden. In einem Test des im Rahmen der Arbeit entwickelten Prototypen brauchten medizinisch nicht geschulte Probanden im Durchschnitt 2:20 min für die Untersuchung eines Datensatzes und die Skizzierung von Größe und Position gefundener Verwirbelungen. Durch den Wegfall der abhängig von der Qualität des Datensatzes oft mehrere Minuten dauernden manuellen Segmentierung sind somit qualitative Untersuchungen in sehr kurzer Zeit möglich.

Unterstützt wird dies durch die Hervorhebung von Features des Blutflusses. Mithilfe des λ_2 -Wirbelkriteriums wurden Verwirbelungen automatisch erkannt und im Blutfluss farblich hervorgehoben. Im Gefäß liegende Wirbelstrukturen können zudem sowohl durch Overlays als auch durch Ausblenden des laminaren Flusses sichtbar gemacht werden. Auch die Hervorhebung anderer Features, wie Flussgeschwindigkeiten oder Rückflüsse, ist möglich. Durch die Hervorhebung der Wirbelstrukturen waren die Teilnehmer des Tests in der Lage, 73.6% der in den Datensätzen vorhandenen Verwirbelungen zu erkennen.

Ein Problem bei der Flussanalyse stellten Datensätze mit geringem Kontrast in Teilen der Gefäße dar. In solchen Fällen wurden diese unvollständig dargestellt und eine Analyse war nur eingeschränkt möglich, da bei einer entsprechenden Anpassung der Visualisierungsparameter auch Bereiche außerhalb der Gefäße angezeigt werden und diese verdecken. Zudem werden in diesem Fall auch Partikel außerhalb der Gefäße

platziert. Eine Visualisierung der Anatomie mittels MIP ist bei solchen Datensätzen möglich, die Flussvisualisierung hingegen problematisch. Dies könnte durch einen intelligenteren Seeding-Algorithmus abgemildert werden, der beispielsweise zusätzlich zur Selektion von Seeding-Voxeln über einen Schwellenwert ein gradientenbasiertes Region Growing anwendet.

Ein weiteres Problem sind Verdeckungen in der Anatomie, wodurch die Sicht auf bestimmte Gefäße eingeschränkt wird. Abhilfe könnte hier ein Clustering-Verfahren schaffen, welches eine automatische, grobe Segmentierung durchführt und somit das Ausblenden einzelner Strukturen erlaubt.

Für die Anpassung der Darstellung reichen im Wesentlichen zwei Parameter (Schwellenwert für die Visualisierung und λ_2 -Schwellenwert) aus. Zusätzliche Funktionen wie Overlays oder das Ausblenden von laminarem Fluss stehen zur Verfügung, sind für eine vollständige Visualisierung jedoch nicht zwingend erforderlich. Die eingeschränkte Darstellung der Anatomie bei kontrastarmen Datensätzen stellt jedoch ein bedeutendes Hindernis dar. Zwar können Visualisierungsparameter wie Schwellenwerte oder Transferfunktionen manuell angepasst werden, allerdings lässt sich bei sehr schwachem Kontrast keine Einstellung finden, die ein akzeptables Ergebnis erzeugt.

Prinzipiell lässt sich das hier vorgestellte Verfahren auch auf andere Blutgefäße anwenden, vorausgesetzt sie sind groß genug, um durch 4D PC-MRI Untersuchungen erfasst zu werden. Allerdings existiert auch hier das Problem der Verdeckung, das bei Gefäßbäumen oder anderen komplexen Topologien stark ausgeprägt sein wird. Zudem muss die Funktion zur Histogrammanalyse und automatischer Festlegung der Visualisierungsparameter gegebenenfalls an die geänderten Anforderungen angepasst werden.

Theoretisch ließen sich noch weitere Features für den Blutfluss berechnen und darstellen oder bestehende Features erweitern. Durch die zusätzliche Betrachtung der curl-Vektoren in einem durch das λ_2 -Wirbelkriterium markierten Bereich könnte beispielsweise die Drehrichtung des Wirbels visualisiert werden. Auch könnten durch eine Kombination des Wirbelkriteriums mit der Flussgeschwindigkeit Fehlerkennungen durch am Ende der Systole abbremsenden Blutfluss vermieden werden. Somit würden Partikel, deren Bewegungsgeschwindigkeit unterhalb eines bestimmten Schwellenwertes liegt, nicht mehr für die Wirbelextraktion betrachtet werden.

Die Integration von Funktionen zur Quantifizierung des Blutflusses, beispielsweise zur Messung von Schlag- oder Rückflussvolumen, wäre ebenfalls prinzipiell möglich. Da jedoch keine Segmentierung zur Verfügung steht, stellt die Positionierung von Messebenen parallel zum Gefäßquerschnitt ein Problem dar. Hierfür müsste eine Gefäßmittellinie sowie der Gefäßdurchmesser automatisch berechnet werden. Alternativ wäre auch eine vollständig manuelle Platzierung sowie Skalierung der Messebene durch den Anwender möglich. Dies würde jedoch komplexe Interaktionen mit der Visualisierung voraussetzen und der Idee der Automatisierung zuwiderlaufen.

Literaturverzeichnis

- [ABP⁺13] ALLEN, B. D. ; BARKER, A. J. ; PAREKH, K. ; SOMMERVILLE, L. C. ; SCHNELL, S. ; JARVIS, K. B. ; CARR, M. ; CARR, J. ; COLLINS, J. ; MARKL, M.: Incorporating time-resolved three-dimensional phase contrast (4D flow) MRI in clinical workflow: initial experiences at a large tertiary care medical center. In: *Resonance* 15 (2013), Nr. 1, S. P32 (zitiert auf Seite 21)
- [AWR09] APITZ, C. ; WEBB, G. D. ; REDINGTON, A. N.: Tetralogy of Fallot. In: *The Lancet* 374 (2009), Nr. 9699, S. 1462–1471. – ISSN 01406736 (zitiert auf Seite 5)
- [BA09] BAILLIARD, F. ; ANDERSON, R. H.: Tetralogy of Fallot. In: *Orphanet journal of rare diseases* 4 (2009), S. 2. – ISSN 1750–1172 (zitiert auf Seite 5)
- [BBV04] BOGREN, H. G. ; BUONOCORE, M. H. ; VALENTE, R. J.: Four-dimensional magnetic resonance velocity mapping of blood flow patterns in the aorta in patients with atherosclerotic coronary artery disease compared to age-matched normal subjects. In: *Journal of magnetic resonance imaging : JMRI* 19 (2004), Nr. 4, S. 417–427. – ISSN 1522–2586 (zitiert auf Seite 5)
- [BHB⁺13] BISSELL, M. M. ; HESS, A. T. ; BIASIOLLI, L. ; GLAZE, S. J. ; LOUDON, M. ; PITCHER, A. ; DAVIS, A. ; PRENDERGAST, B. ; MARKL, M. ; BARKER, A. J. ; NEUBAUER, S. ; MYERSON, S. G.: Aortic dilation in bicuspid aortic valve disease: flow pattern is a major contributor and differs with valve fusion type. In: *Circulation. Cardiovascular imaging* 6 (2013), Nr. 4, S. 499–507. – ISSN 1942–0080 (zitiert auf Seite 5)
- [BMB⁺12] BARKER, A. J. ; MARKL, M. ; BÜRK, J. ; LORENZ, R. ; BOCK, J. ; BAUER, S. ; SCHULZ-MENGER, J. ; KNOBELSDORFF-BRENKENHOFF, F. v.: Bicuspid aortic valve is associated with altered wall shear stress in the ascending aorta. In: *Circulation. Cardiovascular imaging* 5 (2012), Nr. 4, S. 457–466. – ISSN 1942–0080 (zitiert auf Seite 5)
- [BMDH07] BURGER, R. ; MUIGG, P. ; DOLEISCH, H. ; HAUSER, H.: Interactive cross-detector analysis of vortical flow data. In: *Fifth International Conference on Coordinated and Multiple Views in Exploratory Visualization (CMV 2007)*, 2007, S. 98–110 (zitiert auf Seite 29)

- [BPM⁺13] BORN, S. ; PFEIFLE, M. ; MARKL, M. ; GUTBERLET, M. ; SCHEUERMANN, G.: Visual analysis of cardiac 4D MRI blood flow using line predicates. In: *Visualization and Computer Graphics, IEEE Transactions on* 19 (2013), Nr. 6, S. 900–912 (zitiert auf Seite 24)
- [BPS⁺13] BÄCHLER, P. ; PINOCHET, N. ; SOTELO, J. ; CRELIER, G. ; IRARRAZAVAL, P. ; TEJOS, C. ; URIBE, S.: Assessment of normal flow patterns in the pulmonary circulation by using 4D magnetic resonance velocity mapping. In: *Magnetic resonance imaging* 31 (2013), Nr. 2, S. 178–188. – ISSN 1873–5894 (zitiert auf Seite 22)
- [BS95] BANKS, D. C. ; SINGER, B. A.: A Predictor-Corrector Technique for Visualizing Unsteady Flow. In: *IEEE Transactions on Visualization and Computer Graphics* (1995), Nr. 1, S. 151–163. – ISSN 1077–2626 (zitiert auf Seite vi, 28 und 29)
- [CL93] CABRAL, B. ; LEEDOM, L. C.: Imaging vector fields using line integral convolution. In: *Proceedings of the 20th annual conference on Computer graphics and interactive techniques*. 1993, S. 263–270 (zitiert auf Seite vi, 23 und 24)
- [CLT06] CHRISTEN, T. ; LERCH, R. ; TRINDADE, P. T.: Die Aortenklappenstenose beim Erwachsenen: Teil 1. Ätiologie, Pathophysiologie und Diagnose. In: *Schweiz Med Forum* (2006), Nr. 6, S. 626–632 (zitiert auf Seite 5)
- [DCH88] DREBIN, R. A. ; CARPENTER, L. ; HANRAHAN, P.: Volume Rendering. In: *j-COMP-GRAPHICS* 22 (1988), Nr. 4, S. 65–74. – ISSN 0097–8930 (zitiert auf Seite 14)
- [Die14] DIETZ, H. C. ; UNIVERSITY OF WASHINGTON, SEATTLE (Hrsg.): *Marfan Syndrome*. <http://www.ncbi.nlm.nih.gov/books/NBK1335/>. Version: 2014 (zitiert auf Seite 4)
- [DKS⁺08] DYVERFELDT, P. ; KVITTING, J. E. ; SIGFRIDSSON, A. ; ENGVALL, J. ; BOLGER, A. F. ; EBBERS, T.: Assessment of fluctuating velocities in disturbed cardiovascular blood flow: in vivo feasibility of generalized phase-contrast MRI. In: *Journal of magnetic resonance imaging : JMRI* 28 (2008), Nr. 3, S. 655–663. – ISSN 1522–2586 (zitiert auf Seite 1 und 6)
- [EE02] ENGEL, K. ; ERTL, T.: Interactive High-Quality Volume Rendering with Flexible Consumer Graphics Hardware. In: *Eurographics Conference on Visualization* (2002) (zitiert auf Seite v und 15)
- [FBM⁺12] FRYDRYCHOWICZ, A. ; BERGER, A. ; MUNOZ DEL RIO, A. ; RUSSE, M. F. ; BOCK, J. ; HARLOFF, A. ; MARKL, M.: Interdependencies of aortic arch secondary flow patterns, geometry, and age analysed by 4-dimensional phase contrast magnetic resonance imaging at 3 Tesla. In: *European radiology* 22 (2012), Nr. 5, S. 1122–1130. – ISSN 1432–1084 (zitiert auf Seite 21)

- [FMH⁺11] FRYDRYCHOWICZ, A. ; MARKL, M. ; HIRTLE, D. ; HARLOFF, A. ; SCHLENSAK, C. ; GEIGER, J. ; STILLER, B. ; ARNOLD, R.: Aortic Hemodynamics in Patients With and Without Repair of Aortic Coarctation: In Vivo Analysis by 4D Flow-Sensitive Magnetic Resonance Imaging. In: *Investigative Radiology* 46 (2011), Nr. 5 (zitiert auf Seite 22)
- [FSS⁺12] FRANÇOIS, C. J. ; SRINIVASAN, S. ; SCHIEBLER, M. L. ; REEDER, S. B. ; NIESPODZANY, E. ; LANDGRAF, B. R. ; WIEBEN, O. ; FRYDRYCHOWICZ, A.: 4D cardiovascular magnetic resonance velocity mapping of alterations of right heart flow patterns and main pulmonary artery hemodynamics in tetralogy of Fallot. In: *Journal of cardiovascular magnetic resonance : official journal of the Society for Cardiovascular Magnetic Resonance* 14 (2012), S. 16. – ISSN 1532–429X (zitiert auf Seite 22)
- [FZE⁺11] FREDRIKSSON, A. G. ; ZAJAC, J. ; ERIKSSON, J. ; DYVERFELDT, P. ; BOLGER, A. F. ; EBBERS, T. ; CARLHÄLL, C.: 4-D blood flow in the human right ventricle. In: *American journal of physiology. Heart and circulatory physiology* 301 (2011), Nr. 6, S. H2344–50. – ISSN 1522–1539 (zitiert auf Seite 21)
- [Gro10] GROSCH, T.: *Ray Tracing*. Magdeburg, 2010 (Photorealistische Computergrafik) (zitiert auf Seite v, 15 und 16)
- [HFS⁺11] HENNEMUTH, A. ; FRIMAN, O. ; SCHUMANN, C. ; BOCK, J. ; DREXL, J. ; HUELLEBRAND, M. ; MARKL, M. ; PEITGEN, H.-O.: *Fast Interactive Exploration of 4D MRI Flow Data*. 2011 (zitiert auf Seite v, 6 und 7)
- [HGH⁺10] HUMMEL, M. ; GARTH, C. ; HAMANN, B. ; HAGEN, H. ; JOY, K. I.: IRIS: illustrative rendering for integral surfaces. In: *IEEE Transactions on Visualization and Computer Graphics* 16 (2010), Nr. 6, S. 1319–1328. – ISSN 1077–2626 (zitiert auf Seite 25)
- [HHM⁺10] HOPE, M. D. ; HOPE, T. A. ; MEADOWS, A. K. ; ORDOVAS, K. G. ; URBANIA, T. H. ; ALLEY, M. T. ; HIGGINS, C. B.: Bicuspid aortic valve: four-dimensional MR evaluation of ascending aortic systolic flow patterns. In: *Radiology* 255 (2010), Nr. 1, S. 53–61. – ISSN 1527–1315 (zitiert auf Seite 22)
- [HJ05] HANSEN, C. D. ; JOHNSON, C. R.: Detection and visualization of vortices. In: *The Visualization Handbook*. Academic Press, 2005, S. 295–309 (zitiert auf Seite 27)
- [HPT⁺08] HONG, G. ; PEDRIZZETTI, G. ; TONTI, G. ; LI, P. ; WEI, Z. ; KIM, J. K. ; BAWEJA, A. ; LIU, S. ; CHUNG, N. ; HOULE, H. ; NARULA, J. ; VANNAN, M. A.: Characterization and quantification of vortex flow in the human left ventricle by contrast echocardiography using vector particle image velocimetry. In: *JACC. Cardiovascular imaging* 1 (2008), Nr. 6, S. 705–717. – ISSN 1876–7591 (zitiert auf Seite 26)

- [HWHJ99] HECKEL, B. ; WEBER, G. ; HAMANN, B. ; JOY, K. I.: Construction of vector field hierarchies. In: *Proceedings of the conference on Visualization'99: celebrating ten years*, 1999, S. 19–25 (zitiert auf Seite vi und 26)
- [JH94] JEONG, J. ; HUSSAIN, F.: On the identification of a vortex. In: *Journal of fluid mechanics* 285 (1994), S. 69–94 (zitiert auf Seite 28 und 29)
- [Kar09] KARAMALIS, A.: *Einführung in GLSL - OpenGL Shading Language*. <http://campar.in.tum.de/twiki/pub/Chair/TeachingWs09CPP/GLSL.pdf>. Version: 2009 (zitiert auf Seite 10)
- [KD98] KINDLMANN, G. ; DURKIN, J. W.: Semi-automatic generation of transfer functions for direct volume rendering. In: *Proceedings of the 1998 IEEE symposium on Volume visualization*, 1998, S. 79–86 (zitiert auf Seite 19)
- [KG01] KÖNIG, A. H. ; GRÖLLER, E. M.: Mastering Transfer Function Specification by using VolumePro Technology. In: *Spring Conference on Computer Graphics* (2001) (zitiert auf Seite vi und 19)
- [KGP⁺13] KÖHLER, B. ; GASTEIGER, R. ; PREIM, U. ; THEISEL, H. ; GUTBERLET, M. ; PREIM, B.: Semi-automatic vortex extraction in 4D PC-MRI cardiac blood flow data using line predicates. In: *IEEE Transactions on Visualization and Computer Graphics* 19 (2013), Nr. 12, S. 2773–2782. – ISSN 1077–2626 (zitiert auf Seite 4)
- [Kha11] KHAN, H. A.: *Characterization of Flow Patterns in MRI Phase Contrast Data*. Karlsruhe, Germany, Karlsruhe Institute of Technology, Master Thesis, 2011 (zitiert auf Seite vi und 27)
- [Khr15] KHRONOS GROUP: *Compute Shader*. https://www.opengl.org/wiki/Compute_Shader. Version: 2015 (zitiert auf Seite 13)
- [Köh12] KÖHLER, B.: *Wirbelextraktion in 4D PC-MRI Blutflussdaten der Aorta*. Magdeburg, Otto-von-Guericke Universität, Masterarbeit, 2012 (zitiert auf Seite v, vi, 1, 8, 9, 24, 29 und 30)
- [Kub14] KUBISCH, C.: *Order Independent Transparency In OpenGL 4.x*. <http://on-demand.gputechconf.com/gtc/2014/presentations/S4385-order-independent-transparency-opengl.pdf>. Version: 2014 (GPU Technology Conference) (zitiert auf Seite 46)
- [KW03] KRÜGER, J. ; WESTERMANN, R.: Acceleration Techniques for GPU-based Volume Rendering. In: *EEE Visualization 2003* (2003) (zitiert auf Seite v, 16 und 17)
- [LDS90] LEVI, Y. ; DEGANI, D. ; SEGNER, A.: Graphical visualization of vortical flows by means of helicity. In: *AIAA Journal* 28 (1990), Nr. 8, S. 1347–1352. – ISSN 0001–1452 (zitiert auf Seite 27)

- [LHM⁺05] LANKHAAR, J. ; HOFMAN, M. ; MARCUS, J. T. ; ZWANENBURG, J. ; FAES, T. ; VONK-NOORDEGRAAF, A.: Correction of phase offset errors in main pulmonary artery flow quantification. In: *Journal of magnetic resonance imaging : JMRI* 22 (2005), Nr. 1, S. 73–79. – ISSN 1522–2586 (zitiert auf Seite 6)
- [LKR⁺04] LI, A. E. ; KAMEL, I. ; RANDO, F. ; ANDERSON, M. ; KUMBASAR, B. ; LIMA, J. ; BLUEMKE, D. A.: Using MRI to assess aortic wall thickness in the multiethnic study of atherosclerosis: distribution by race, sex, and age. In: *AJR. American journal of roentgenology* 182 (2004), Nr. 3, S. 593–597. – ISSN 0361–803X (zitiert auf Seite 47)
- [LMLG02] LOTZ, J. ; MEIER, C. ; LEPPERT, A. ; GALANSKI, M.: Cardiovascular flow measurement with phase-contrast MR imaging: basic facts and implementation. In: *Radiographics : a review publication of the Radiological Society of North America, Inc* 22 (2002), Nr. 3, S. 651–671. – ISSN 0271–5333 (zitiert auf Seite 6)
- [MGH⁺11] MARKL, M. ; GEIGER, J. ; HERZER, L. ; STILLER, B. ; ARNOLD, R.: 3D Aortic Blood Flow in Patients with Marfan Syndrome: Changes in Hemodynamics and Correlation with Aortic Geometry. In: *Proc. Intl. Soc. Mag. Reson. Med* 19 (2011) (zitiert auf Seite 22)
- [MSL⁺13] MEIERHOFER, C. ; SCHNEIDER, E. P. ; LYKO, C. ; HUTTER, A. ; MARTINOFF, S. ; MARKL, M. ; HAGER, A. ; HESS, J. ; STERN, H. ; FRATZ, S.: Wall shear stress and flow patterns in the ascending aorta in patients with bicuspid aortic valves differ significantly from tricuspid aortic valves: a prospective study. In: *European heart journal cardiovascular Imaging* 14 (2013), Nr. 8, S. 797–804. – ISSN 2047–2412 (zitiert auf Seite 22)
- [MTHG03] MATTAUSCH, O. ; THEUSSL, T. ; HAUSER, H. ; GRÖLLER, E.: Strategies for interactive exploration of 3D flow using evenly-spaced illuminated streamlines. In: *Proceedings of the 19th spring conference on Computer graphics*. 2003, S. 213–222 (zitiert auf Seite vi, 24 und 25)
- [NVI14] NVIDIA: *Programming Guide :: CUDA Toolkit Documentation*. <http://docs.nvidia.com/cuda/cuda-c-programming-guide/>. Version: 2014 (zitiert auf Seite 13)
- [Nyq28] NYQUIST, H.: Certain Topics in Telegraph Transmission Theory. In: *Transactions of the American Institute of Electrical Engineers* (1928), Nr. 47, S. 614–644 (zitiert auf Seite 17)
- [PVH⁺03] POST, F. H. ; VROLIJK, B. ; HAUSER, H. ; LARAMEE, R. S. ; DOLEISCH, H.: The state of the art in flow visualisation: Feature extraction and tracking. In: *Computer Graphics Forum* Bd. 22, 2003, S. 775–792 (zitiert auf Seite 23, 24 und 25)

- [Rák10] RÁKOS, D.: *History of hardware tessellation*. <http://rastergrid.com/blog/2010/09/history-of-hardware-tessellation/>. Version: 2010 (zitiert auf Seite v und 11)
- [Rot00] ROTH, M.: *Automatic Extraction of Vortex Core Lines and Other Line-Type Features for Scientific Visualization*, ETH Zürich, Dissertation, 2000 (zitiert auf Seite vi, 28 und 29)
- [Sch10] SCHLECHTWEG, S.: *Computergraphik Grundlagen: VIII. Beleuchtung und Shading*. <http://www.mttcs.org/Skripte/Pra/Material/vorlesung7.pdf>. Version: 2010 (zitiert auf Seite v und 11)
- [SFH⁺10] STALDER, A. F. ; FRYDRYCHOWICZ, A. ; HARLOFF, A. ; YANG, Q. ; BOCK, J. ; HENNIG, J. ; LI, K. C. ; MARKL, M.: Vortex core detection and visualization using 4D flow-sensitive MRI. In: *Proc. Intl. Soc. Mag. Reson. Med* Bd. 18, 2010, S. 3708 (zitiert auf Seite vi, 25 und 26)
- [SGGJ13] STALDER, A. F. ; GULSUN, M. A. ; GREISER, A. ; JOLLY, M.: Fully automatic visualization of 4D Flow data. In: *Proc. Intl. Soc. Mag. Reson. Med* 21 (2013), S. 1434 (zitiert auf Seite 2)
- [SJWS08] SALZBRUNN, T. ; JÄNICKE, H. ; WISCHGOLL, T. ; SCHEUERMANN, G.: The State of the Art in Flow Visualization: Partition-Based Techniques. In: *SimVis*, 2008, S. 75–92 (zitiert auf Seite 25)
- [SKGM03] SHIN, H. ; KING, B. ; GALANSKI, M. ; MATTHIES, H. K.: Development of an intuitive graphical user interface for volume rendering of multi-detector CT data. In: *International Congress Series* 1256 (2003), S. 265–269. – ISSN 05315131 (zitiert auf Seite vi und 19)
- [SVSG06] SEREDA, P. ; VILANOVA, B. A. ; SERLIE, I. ; GERRITSEN, F. A.: Visualization of boundaries in volumetric data sets using LH histograms. In: *IEEE Transactions on Visualization and Computer Graphics* 12 (2006), Nr. 2, S. 208–218. – ISSN 1077–2626 (zitiert auf Seite vi, 19 und 20)
- [Van91] VAN WIJK, J. J.: Spot noise texture synthesis for data visualization. In: *ACM Siggraph Computer Graphics* 25 (1991), Nr. 4, S. 309–318 (zitiert auf Seite 23)
- [vOB⁺10a] VAN PELT, R. ; OLIVÁN BESCÓS, J. ; BREEUWER, M. ; CLOUGH, R. E. ; GRÖLLER, M. E. ; TER HAAR ROMENIJ, B. ; VILANOVA, A.: Exploration of 4D MRI blood flow using stylistic visualization. In: *IEEE Transactions on Visualization and Computer Graphics* 16 (2010), Nr. 6, S. 1339–1347. – ISSN 1077–2626 (zitiert auf Seite 23)
- [vOB⁺10b] VAN PELT, R. ; OLIVÁN BESCÓS, J. ; BREEUWER, M. ; GRÖLLER, M. E. ; TER HAAR ROMENY, B. ; VILANOVA, A.: Exploring Hemodynamics by Raycasting 4D MRI Flow. (2010) (zitiert auf Seite 24)
- [WNR⁺08] WEIGANG, E. ; NIENABER, C. A. ; REHDESS, T. C. ; INCE, H. ; VAHL, C. ; BEYERSDORF, F.: Management of patients with aortic dissection.

- In: *Deutsches Ärzteblatt international* 105 (2008), Nr. 38, S. 639–645. – ISSN 1866–0452 (zitiert auf Seite 4)
- [YHGT10] YANG, J. C. ; HENSLEY, J. ; GRÜN, H. ; THIBIEROZ, N.: Real-Time Concurrent Linked List Construction on the GPU. In: *Computer Graphics Forum* 29 (2010), Nr. 4, S. 1297–1304. <http://dx.doi.org/10.1111/j.1467-8659.2010.01725.x>. – DOI 10.1111/j.1467–8659.2010.01725.x. – ISSN 01677055 (zitiert auf Seite 46)
- [Zoh11] ZOHAR, R.: *Optimizing DirectX* 8.0 Vertex Shaders*. <https://software.intel.com/en-us/articles/optimizing-directx-80-vertex-shaders>. Version: 2011 (zitiert auf Seite 10)
- [ZSH96] ZOCKLER, M. ; STALLING, D. ; HEGE, H.: Interactive visualization of 3D-vector fields using illuminated stream lines. In: *Visualization'96. Proceedings*. 1996, S. 107–113 (zitiert auf Seite 24)

Hiermit erkläre ich, Benjamin Behrendt, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe.

Benjamin Behrendt

Magdeburg, den 17. April 2015